

**A METHODOLOGY FOR THE VIRTUAL QUANTIFICATION OF OPERATOR
WORKLOADS IN SUPPORT OF PARTIALLY-AUTOMATED SYSTEM
ANALYSIS**

A Dissertation
Presented to
The Academic Faculty

By

Matthew A. C. Guckenberger

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Aerospace Engineering

Georgia Institute of Technology

August 2021

Copyright © Matthew A. C. Guckenberger 2021

**A METHODOLOGY FOR THE VIRTUAL QUANTIFICATION OF OPERATOR
WORKLOADS IN SUPPORT OF PARTIALLY-AUTOMATED SYSTEM
ANALYSIS**

Approved by:

Professor Dimitri Mavris, Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Professor Daniel Schrage
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Alicia Sudol
School of Aerospace Engineering
Georgia Institute of Technology

Professor Mariel Borowitz
School of International Affairs
Georgia Institute of Technology

Dr. Jean Charles Domercant
Electronic Systems Laboratory
Georgia Tech Research Institute

Date Approved: July 07, 2021

The airplane stays up because it doesn't have the time to fall

Orville Wright

ACKNOWLEDGEMENTS

This has been quite the journey and I would like to take a moment to thank those who have helped me along the way. First, I would like to express my appreciation to my committee members: Prof. Dimitri Mavris, Prof. Daniel Schrage, Prof. Mariel Borowitz, Dr. Charles Domercant, and Dr. Alicia Sudol. Your time and feedback have helped shape this work and improved me as a scholar. I would especially like to thank Dr. Alicia Sudol and Prof. Dimitri Mavris. Dr. Sudol has been my research engineer for most of my time at Georgia Tech, and provided me with a wealth of growth opportunities. She has spent countless hours providing feedback on my research and ensuring that I never felt alone when stress got high. Prof. Mavris, my advisor, has given me the time and support to grow. He continually shares pieces of knowledge, many of which I expect to be useful for the rest of my life. My time at Georgia Tech and the Aerospace Systems Design Laboratory (ASDL) would not have been possible or as rewarding without his tireless effort.

I would also like to express my appreciation to my friends and family. I would not be here today without the love and support from my parents, Dave and Debra. My brothers, John and Alex, always ensured that I had someone to talk with and a little comic relief. I would also like to thank Brooke for her love and patience during this process. She has been the light at the end of every long day. To my friends, I cannot list everyone who has impacted me during my time at Tech and cannot even pick a few because I owe each of my gratitude. From the friendly catch-ups to the needed time away from research, all of you contributed to my journey and I want to thank you.

I could not be where I am today without such a supportive network. This PhD process and my life have been made better by having each of you in them. Thank you.

TABLE OF CONTENTS

Acknowledgments	iv
List of Tables	viii
List of Figures	x
Summary	xvii
Chapter 1: Introduction and Motivation	1
Chapter 2: Technical Approach	16
2.1 Background	16
2.1.1 Systems Engineering and System Development	16
2.1.2 Operations Research	27
2.2 Gaps	43
2.3 Research Objective	47
2.4 Research Problem	48
2.4.1 Formulation	49
2.4.2 Analysis	54
2.4.3 Model Solution	80
2.4.4 Technology Trade Space Evaluation	82

2.4.5	Interpretation	91
2.4.6	Demonstration	95
Chapter 3: Benchmarking Methodology using Automotive Example		98
3.1	Experiment 1 - Benchmarking	98
3.1.1	Experiment 1 Setup	100
3.1.2	Experiment 1 Results	116
3.2	Experiment 2 - Automation Modifications	132
3.2.1	Experiment 2 Setup	134
3.2.2	Experiment 2 Results	137
3.3	Experiment 3 - Automation Assessments	144
3.3.1	Experiment 3 Setup	145
3.3.2	Experiment 3 Results	149
Chapter 4: Demonstration of Methodology to Aerospace Application		157
4.1	Experiment 4 Setup	158
4.1.1	Operations Model	159
4.1.2	Dynamic Task List	162
4.1.3	Sensory Model	165
4.1.4	Workload Model	168
4.1.5	Technology Integration	169
4.2	Experiment 4 Results	174
4.2.1	Trade Space	180

Chapter 5: Discussion and Conclusion	184
5.1 Contributions	193
5.2 Future Work	195
Appendix A: Summary of Research Questions	198
Appendix B: Summary of Experiments	200

LIST OF TABLES

1.1	Levels of Automation in Man-Computer Decision-Making for a Single Elemental Decisive Step [98].	2
1.2	Ten Challenges in Human-agent Teaming [59].	9
1.3	Trades studies which are typically conducted during the aircraft design process [87].	15
2.1	Qualities of a Good Requirement according to the <i>Handbook of Systems Engineering and Management</i> [6].	20
2.2	Characteristics of Good Requirements according to the <i>Department of Defense Handbook</i> [22].	20
2.3	Characterizing Features of Epistemic and Aleatory Uncertainty [35].	41
2.4	Taxonomy for operator workload drivers, primary class and subclass [52].	43
2.5	Self-reported average workload ratings by condition [27].	51
2.6	Definitions for the six rating scales used by NASA's Task Load Index (TLX) [47].	69
2.7	A summary of Research Question 1 with its related sub-questions and assertions/hypothesis	74
2.8	Environment settings for the agent-based model. From Rahman et. al's human-in-the-loop study [86].	82
2.9	A summary of Experiment 1.0 which aims to test the hypothesis and sub-hypothesis of Research Question 1.0	82
2.10	A summary of Research Question 2 with its related sub-questions and assertions/hypothesis	88

2.11	A summary of Experiment 2.0 which aims to test the hypothesis and sub-hypothesis of Research Question 2.0	91
2.12	A summary of Research Question 3	94
2.13	A summary of Experiment 3.0 which aims to test the hypothesis of Research Question 3.0	95
2.14	A summary of Experiment 4.0 which aims to test Hypothesis 4.0 and validate the overall research objective	97
A.1	A summary of research questions with related sub-questions and assertions/hypotheses	198
A.2	A summary of research questions with related sub-questions and assertions/hypotheses (cont.)	199
B.1	A summary of experiments	200
B.2	A summary of experiments (cont.)	201

LIST OF FIGURES

1.1	Relationship between autonomy and automation [36].	3
1.2	The six pillars for engineering automation on aviation systems	4
1.3	RAND report showing the number of miles required to demonstrate an autonomous vehicles functionality under different level of statistical confidence [54].	6
1.4	SAE Standard for Levels of Driving Automation [89].	7
1.5	The frequency which operators of different skill levels used automated data and raw data in a NUWC study [56].	10
1.6	Results from a study assessing performance under different levels of autonomy for a gunner also controlling and Unmanned Ground Vehicle (UGV) [12]	11
1.7	NTSB graph showing the amount of operator control before the 2018 Tesla crash [109].	12
1.8	Hebb/Yerkes-Dodson Law relating arousal to performance for difficult and simple tasks [58].	14
2.1	The Systems Engineering Vee Model [25]	18
2.2	The relationship of design freedom, knowledge, and cost committed throughout the acquisition timeline [71]	19
2.3	Department of Defense life cycle acquisition process with reference to costs committed and program expenditures [83]	22
2.4	The Concept Development Systems Engineering (CDSE) Process [95]	23
2.5	Integrated Product/Process Development (IPPD) [94]	24

2.6	Technology Identification, Evaluation, and Selection (TIES) for Systems (just using the top path) and System of Systems (the top and bottom with iteration) [73]	26
2.7	The 7-step Operations Research process showing the typical three-phase breakdown. Adapted from: [121, 10]	31
2.8	Levels of military modeling and simulation activities. Adapted from: [8] . .	33
2.9	Modeling and Simulation Spectrum for Engineering [49]	35
2.10	General Taxonomy of an Intelligent Software Agent [112]	36
2.11	A simplified, high level representation showing the primary components of a finite state machine for controlling Ms Pac-Man [118]	38
2.12	A simplified behavior tree for controlling Ms Pac-Man's seek pellets behavior [118]	39
2.13	The three pieces that must be added to the operations model to reduce uncertainty around operator workload	45
2.14	Overview of the four stages to create and evaluate a methodology that decreases the uncertainty pertaining to operator workload in partially-automated system analysis	49
2.15	Results from Edwards et. al showing the time taken by an air traffic controller to detect conflicts under different levels of automation (conditions detailed in Table 2.4) [27]	51
2.16	Results from Schneider et. al showing modeled workload levels for and ISR mission with and without multi-aircraft Control (MAC) [93]	53
2.17	The notional relationship between the fields of system design, operations research, and cognitive analysis to provide technology assessments	57
2.18	Flow diagram representing the steps to conduct a hierarchical task analysis [102]	59
2.19	Procedure breakdown for an aircraft performing a go-around using a hierarchical task analysis [64]	60
2.20	NHTSA operational event sequence diagram (OESD) for changing lanes [62]	61

2.20 NHTSA operational event sequence diagram (OESD) for changing lanes (cont.) [62]	62
2.20 NHTSA operational event sequence diagram (OESD) for changing lanes (cont.) [62]	63
2.21 Endsley’s model for situation awareness in dynamic decision making [31] .	65
2.22 Dependence between environment complexity, operator workload, and sys- tem performance [21]	66
2.23 Key elements and general flow of an IMPRINT Pro model [44]	71
2.24 Sample network diagram for tasks involved in a tank tactical march back to an assembly area [44]	72
2.25 Mapping between taxons and a maintenance task within IMPRINT [44] . .	72
2.26 Simulation tools approximate market shares based 2016 study on LinkedIn user groups and publications at WinterSim 2016 [39]	75
2.27 Example of a Simio agent-based model and visualization utilized in the modeling of an airport terminal sales [2]	75
2.28 Example of a NetLogo agent-based model and visualization utilized in the modeling of a traffic grid [115]	76
2.29 Example of an AnyLogic agent-based model and visualization showing fork lift movement to load machinery in a factory model [38]	76
2.30 Example of an AFSIM agent-based model and visualization depicting an air combat scenario [111]	77
2.31 Matrix of alternatives describing the four chosen agent-based simulation packages with evaluation metrics [38, 115, 17, 82]	79
2.32 NASA TLX workload ratings from a study assessing driver workload un- der different environmental conditions (highway [simple], rural [moderate], and city [complex]) [86]	83
2.32 NASA TLX workload ratings from a study assessing driver workload un- der different environmental conditions (highway [simple], rural [moderate], and city [complex]) (cont.) [86]	84

2.33	Overview of functionality of driving automation technologies integrated in the agent-based model. Figures from:	90
2.34	Notional graph showing cognitive workload over time for a graphical versus voice-based user interface [114]	93
2.35	NTSB graph showing the amount of operator control before the 2018 Tesla crash [109].	93
3.1	Summary of research questions, hypotheses, and resulting experiments. . .	99
3.2	Layout of downtown Atlanta city roads in AFSIM	102
3.3	Representation of the behavior tree for driver agent showing the four primary branches: navigation, road monitoring, changing lanes, and speed control	106
3.4	The six information channels for the sensory model: left side, right side, left mirror, right mirror, near front, and far front. The behavior tree helps determine which channels are active.	109
3.5	The two active information channels for general driving (near forward and far forward). The near front represents eye scans left to right, while far front the current lane of traffic further out for speed control.	111
3.6	Image depicting the lane evaluation behavior. The driver observes a slow car 200ft ahead in its current lane and has a desire to merge into the left lane. The left side channel is active, shown in green, while the near forward channel is inactive. The far forward channel is also active during this behavior, but is not shown for clarity.	112
3.7	Workload metrics for a highway scenario with 8,000 vehicles. Flow rate is 1,000 cars/ln/hr for the two lane highway, 900 cars/ln/hr for medium density and 1100 cars/ln/hr for high density. Distance traveled was 10.7mi (desired 11.67mi) – 92%	114
3.8	The time spent at each mental workload level for the highway scenario in Figure 3.7 (represented by the number of cars in memory). The time based average for mental demand is 6.55 cars in memory, and the temporal demand is 0.86 [1/sec] (based on the inverse of the average time for changes in mental workload)	115

3.9	The times to change mental workload (the pace of task changes) and the related temporal demands for the highway scenario. The average temporal demand is 0.86 [1/sec] (based on the inverse of the average time for changes in mental workload)	116
3.10	The actions taken by the operator throughout the operational scenario. The same 8,000 car scenario has been used. The width of the spikes demonstrates the duration of the action.	118
3.11	Workload metrics for a city scenario with 2,000 vehicles. Flow rate is 1,000 cars/ln/hr throughout the road network. Distance traveled was 2.63mi (desired 5.00mi) – 53%	122
3.12	The time spent at each mental workload level for the city scenario in Figure 3.11 (represented by the number of cars in memory). The time based average for mental demand is 6.19 cars in memory, and the temporal demand is 2.24 [1/sec] (based on the inverse of the average time for changes in mental workload)	123
3.13	The times to change mental workload (the pace of task changes) and the related temporal demands for the city scenario. The average temporal demand is 2.24 [1/sec] (based on the inverse of the average time for changes in mental workload)	124
3.14	Comparison between experimental and virtual NASA TLX Workload Variations.	125
3.15	Workload variations for changes in the baseline highway flow rate	130
3.16	Mental demand variations for the highway baseline case and two stochastic runs. The stochastic-1 has an average mental demand of 5.03 cars in memory, average temporal demand of 0.72, and a performance of 89%. The stochastic-2 has an average mental demand of 6.57 cars in memory, average temporal demand of 0.88, and a performance of 89%.	131
3.17	Normalized comparison between the human-based study done by Rahman et al.[86], the baseline highway case, and two stochastic variations of the baseline run. Normalization is done using the city workload data.	132
3.18	”Driver” behavior tree for adaptive cruise control (ACC). ACC monitoring has been added to the monitor branch with a speedometer check, and ACC correction is added to the change lanes action based on last remembered speed. The speed control branch is removed from the “driver” and controlled by the car.	135

3.19	Depiction of sensor coverage currently utilized by a highway autopilot system compared with the sensor coverage modeled. The sensor coverage modeled matches the ranges utilized in Experiment 1 but also shows similar coverage to the autopilot	136
3.20	The actions taken by the operator while utilizing adaptive cruise control (ACC). ACC causes a replacement of speed increases and decreases with speedometer checks. The speedometer checks are based on a lognormal distribution with a mean of 30 seconds. The same baseline, 8,000 car scenario has been used. The width of the spikes demonstrates the duration of the action.	139
3.21	The actions taken by the automation while utilizing highway autopilot. The same baseline, 8,000 car scenario has been used. The width of the spikes demonstrates the duration of the action.	141
3.22	Normalized comparison of the operator's workload categories and actions with ACC integration. The specified mean time between speedometer checks captures the trust that the operator maintains in ACC (lower trust results in more frequent speedometer checks).	143
3.23	Example of a design trade space and Pareto frontier. Each circle is centered at an architecture's notional cost-benefit with the diameter representing design risk (uncertainty in the cost-benefit analysis).	146
3.24	The operator's mind-wandering is impacted by variations in workload and time performing the same task [122]	148
3.25	Scatterplot matrix summarizing the workload, action, and awareness results	153
3.26	Example driving automation technology trade space showing the dimension of mental demand versus performance. Mental demand should be minimized, while performance should be maximized.	154
3.27	Example driving automation technology trade space showing the dimension of mental demand versus performance. Mental demand should be minimized, while performance should be maximized.	155
4.1	Operations model in AFSIM showing the layout of the red and blue forces. .	161
4.2	Basic hierarchical task analysis for performing an ISR mission in contested airspace	163

4.3	Representation of the behavior tree for the pilot agent created from the HTA breakdown	165
4.4	Blue aircraft sensory model showing the two-dimensional range for the radar and pilot's vision. The aircraft's current location allows one red fighter to be detected.	167
4.5	A comparison of radar detection ranges. The original range is for the piloted aircraft, while the smaller range is used for the UCAV due to its assumed smaller RCS.	171
4.6	Modified behavior tree for a piloted aircraft with a taskable UCAV	172
4.7	Visualization of operational environment after the inclusion of automation technology (UCAV). The thin blue lines show the blue aircraft routes throughout the mission.	173
4.8	Summary of pilot actions while performing ISR mission solo.	175
4.9	Pilot turn rates while performing the ISR mission solo. The average magnitude is shown as the dashed line at 1.03 degrees per second.	176
4.10	Summary of pilot actions while performing ISR mission jointly with UCAV (RCS is half of piloted aircraft).	177
4.11	Pilot turn rates while performing the ISR mission with the UCAV (RCS is half of piloted aircraft). The average magnitude is shown as the dashed line at 0.56 degrees per second.	178
4.12	Summary of pilot actions while performing ISR mission jointly with UCAV (RCS is equal to the piloted aircraft).	179
4.13	Normalized comparison of average workload metrics (normalized by the results of the solo case).	181
4.14	Normalized comparison of average workload metrics (normalized by the results of the solo case).	182
5.1	Overview of model components in methodology	186
5.2	Process for capturing the role of the operator in a virtual environment to enable automation technology assessments	190

SUMMARY

As systems move through the levels of automation on the way to autonomy, there is a growing and unsustainable amount of uncertainty within the early development process technology evaluations. The integration of automation technology changes the role and awareness of the operator, thus it has a direct impact on the effectiveness of the system. Therefore the effect should be captured during the performance evaluation and requirements definition phases. Automation failures can be seen when looking at the Tesla, which requires little awareness and tasking from the operator, so when the operator is expected to override the automation (as the active manager) it often results in system failure. Similar studies have been done on air traffic controllers which tie together automation level, workload, and performance, to show that maximum automation did not result in maximum system performance. The Department of Defense's trend towards manned-unmanned aerial systems, such as the Boeing Airpower Teaming System which had its first flight in early 2021, will require decreased uncertainty around the operator's effect on system performance. A typical performance evaluation is done through a combination of the systems design and the operations research (OR) fields, however, this leaves a high amount of uncertainty surrounding the operator's awareness and workload throughout the mission. This work captures the operator's workload as a dynamic output throughout the OR evaluation process, through the better capturing of operator awareness and tasking. Moving the workload from a static input to a dynamic output and associating that with system performance provides decision makers a metric by which to assess automation technology in an operational environment. The methodology is demonstrated using two case studies, an assessment of driving technologies and an ISR mission utilizing an unmanned aerial vehicle. The driving case study demonstrates the steps of the methodology and is used for benchmarking. The virtual results showed the same trends and similar normalized differences to experimental data found in literature. The ISR mission demonstrates the utility of

the methodology in the aerospace domain, specifically assessing manned-unmanned teaming. The inclusion of the UCAV reduced the pilot's workload and increased performance, however, the magnitude of the performance increase is dependent on the UCAV's radar signature. The ISR mission also highlighted a possible challenge in managing the UCAV when the piloted aircraft is busy. Overall, the new methodology creates agents that separated the implicit team between the operator and the technology. This separation enables a new dimension of the tradespace focused on the operator's workload and awareness, which is a key enabler for assessing automation technologies early in the design cycle.

CHAPTER 1

INTRODUCTION AND MOTIVATION

The integration of automation technology in vehicle systems and systems of systems has been continually changing the role of the operator and the associated workload. This chapter provides an introduction to the paradigm shift that is driving these changes, and the resulting problems that arise. Over the past decade, automation in vehicles has slowly grown, primarily in the automotive industry, but this is the leading edge of technology integration. An ANSYS White Paper from 2017 emphasizes the disruption expected across industries by autonomous systems: “Autonomous vehicles are threatening to disrupt the automotive, aerospace and industrial equipment industries with the emergence of self-driving cars, drones and mobile autonomous robots. They promise to drastically reduce accidents, minimize congestion, bring mobility to the immobile and perform mundane or hazardous tasks in a fraction of the time required by human-controlled vehicles” [34]. Autonomy is achieved when a system is capable of learning in a dynamic environment and reacting through self-governance and self-directed behavior [36]. Although autonomy has many promises, it will take time and iterative development to realize the goal of full autonomy.

The race towards full system autonomy is on, however, there are intermediate steps in current system design where the operator and technology are being paired through automation technology with the goal of enhancing overall system performance. For the purposes of this paper, automation will use the following definition: “Automation is the use of control systems and information technologies to reduce the need for human work in the production of goods and services” [14]. The variability in automation is best shown using the concept of *Levels of Autonomy (LOA)*. The most cited scale for LOA was done in 1978 by Tom Sheridan and is summarized in Table 1.2 [98]. It shows how automation can take many forms, breaking it into distinct levels, until full autonomy can be achieved. The relation-

Table 1.1: Levels of Automation in Man-Computer Decision-Making for a Single Elemental Decisive Step [98].

Level of Automation	Description of Interaction
1	Human does the whole job up to the point of turning it over to the computer to implement
2	Computer helps by determining the options
3	Computer helps determine options and suggests one, which human needs not follow
4	Computer selects action and human may or may not do it
5	Computer selects action and implements it if human approves
6	Computer selects action, informs human in plenty of time to stop it
7	Computer does the whole job and necessarily tells human what it did
8	Computer does whole job and tells human what it did only if human explicitly asks
9	Computer does whole job and tells human what it did and it, the computer, decides he should be told
10	Computer does whole job if it decides it should be done, and if so tells human, if it decides he should be told

ship between automation and autonomy is shown in Figure 1.1. Integration of automation started with the goal of reducing reaction times and enabling better manning, but as the scope expands, increased and more capable automation must be added to the system until full autonomy is realized.

The trend towards autonomy is being realized in the aerospace and defense industries. As an example, the Boeing Airpower Teaming System pairs unmanned aircraft with a manned aircraft to achieve increased operational effectiveness [18]. The system of systems had its first flight in 2021, and pairs drones flying in formation with a manned aircraft to provide enhanced defensive capabilities. It allows countries which do not have expansive air force resources (pilots or primary aircraft) to achieve increased air power [18]. These systems of systems are not intended to achieve autonomy, rather the automation from the drones is meant to create a force multiplier. The trend towards these automated systems has necessitated the creation of development principles (pillars) within the aerospace field.

Paradigm shift Automation to Autonomy

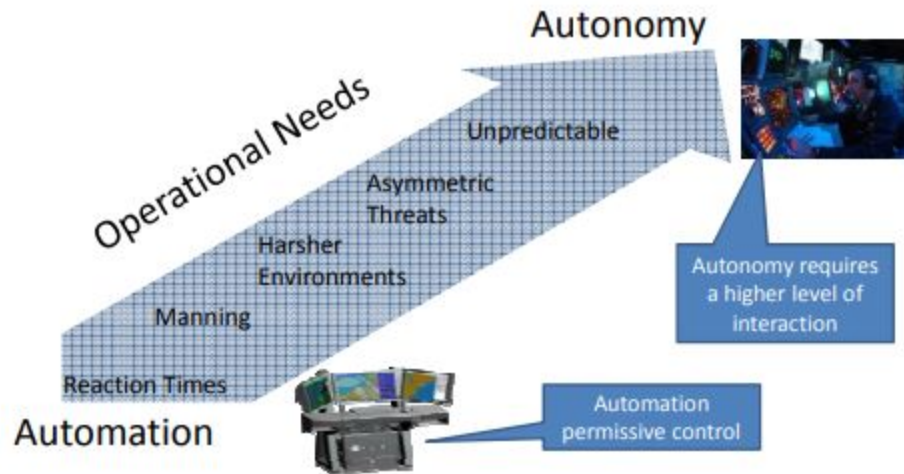


Figure 1.1: Relationship between autonomy and automation [36].

The “Developmental Pillars Of Increased Autonomy For Aircraft Systems” was published in 2020 and is a collaboration between multiple aerospace focused technical committees within ASTM International [23]. The pillars describe characteristics of complex systems that must be evaluated when engineering increasing automation for aviation. The report describes six pillars that constitute the foundational knowledge for automation and autonomy in aviation; the six pillars are shown in Figure 1.2. Development Assurance pertains to the techniques utilized to maintain a level of safety for the complex systems. Modularity and Partitioning ensures the system’s components can be developed and analyzed separate from other unrelated functions. Operational Considerations and the Human Role focuses on assessing automation systems in an operational context, capturing the role of the operator during different stages of automation assistance. Dynamic Consistency Checking monitors the sensor data and checks logical consistency based on established logical principles. Fail Functional Design pertains to design approaches that ensure, in the event of failure, the system continues to function. Finally, Run-Time Assurance is continuous safety checks on the system’s functions and triggering of appropriate recovery behaviors if necessary. Although all six pillars are necessary for automation in aviation

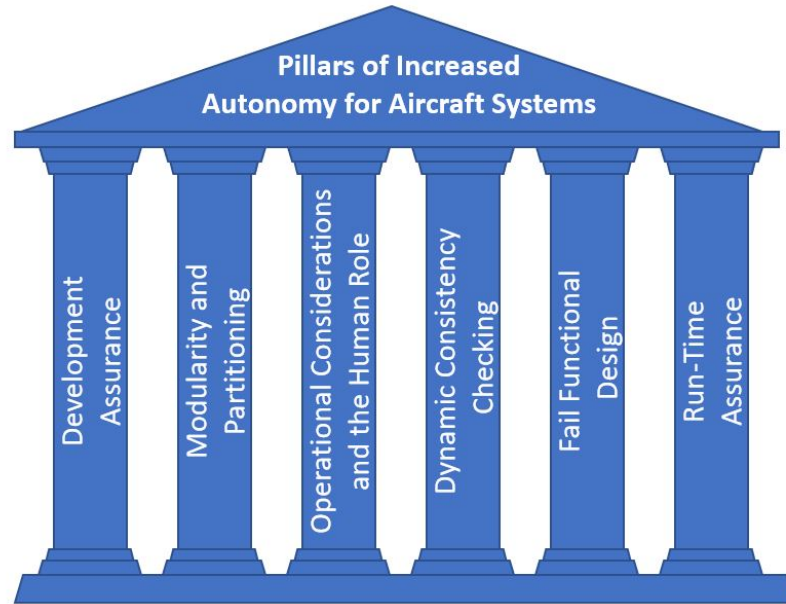


Figure 1.2: The six pillars for engineering automation on aviation systems

systems, this research focuses on Operational Considerations and the Human Role.

Automation in Aviation summarizes the challenges associated with the operator's role and technology integration, "automation has solved old problems but ultimately caused new and different types of accidents" [14]. It discusses how the loss of control for an aircraft has again become a major cause of aviation accidents, however, the driving factors are much different than during the Fifties. During the early days, operator performance was impaired by "under-redundancy". Pilot performance was decreased due to an insufficient level of aids for avoiding fatigue, distraction, workload, and stress. However, many experts point to "over-redundancy" as a current cause of accidents. Increasing automation has put the pilot out-of-the-loop. This causes the pilots to have reduced situational awareness, automation complacency or over-confidence, and over time a loss of skills due to lack of manual flight [14]. From failures such as these, the Operational Considerations and the Human Role Pillar describes three essential requirements for designing future automation: 1) a clearly defined function, 2) a clear operational context of when that function is to be used, and 3) a clear delineation of the expected roles between the human and the automation, considering, both normal and contingency operations [23]. When designing a system it is imperative to

understand the interaction between the automation and the operator.

The level of safety required for aviation applications slows the adoption of emergent technology reducing the number of automation technologies currently fielded, and the proprietary nature of the aerospace industry leaves automation focused aviation data less prevalent. Therefore, to better understand the challenges faced when including automation, it is important to look at other industries. Although aviation and automotive systems are very different, some of the challenges created by the introduction of automation are similar with both industries highlighting the need to understand the role of the operator [23, 89]. The trend towards autonomy is clearly seen in the automotive industry with major players ranging from car manufacturers such as Tesla and Cadillac to service providers such as Uber and Waymo [55]. The industry has already introduced automation features such as automatic emergency braking, lane keeping, and more recently mostly automatic driving. However, the role of the operator or driver remains critical because currently none of the automation technologies have allowed complete autonomy, whereby the technology is capable of adapting to any situation independently [55]. Edge cases, where the system is unaware of the next appropriate action, must be passed to the operator for a decision. Both the aviation and aerospace industries will require a considerable amount of time to demonstrate the safe performance of the machine learning and artificial intelligence algorithms that control the system. A RAND report, shown in Figure 1.3, looked at the number of years to ensure an acceptable failure rate for autonomous vehicles under appropriate statistical confidence levels [54]. This report shows the operator and automation pairing will proceed until millions of miles can be run to validate system functionality. There have been efforts to increase testing rates through virtual testing [34], however, this has yet to be implemented and will pair with operator-automation testing not replace it.

With the automotive industry’s active pursuit of autonomy, the Society of Automotive Engineers (SAE International) put together standard J3016 “Levels of Driving Automation”, the most-cited reference regarding automated-vehicle capabilities [89]. This stan-

Statistical Question	Benchmark Failure Rate			
	How many miles (years ^a) would autonomous vehicles have to be driven...	(A) 1.09 fatalities per 100 million miles?	(B) 77 reported injuries per 100 million miles?	(C) 190 reported crashes per 100 million miles?
	(1) without failure to demonstrate with 95% confidence that their failure rate is at most...	275 million miles (12.5 years)	3.9 million miles (2 months)	1.6 million miles (1 month)
	(2) to demonstrate with 95% confidence their failure rate to within 20% of the true rate of...	8.8 billion miles (400 years)	125 million miles (5.7 years)	51 million miles (2.3 years)
	(3) to demonstrate with 95% confidence and 80% power that their failure rate is 20% better than the human driver failure rate of...	11 billion miles (500 years)	161 million miles (7.3 years)	65 million miles (3 years)

^a We assess the time it would take to complete the requisite miles with a fleet of 100 autonomous vehicles (larger than any known existing fleet) driving 24 hours a day, 365 days a year, at an average speed of 25 miles per hour.

Figure 1.3: RAND report showing the number of miles required to demonstrate an autonomous vehicles functionality under different level of statistical confidence [54].

dard has been captured in a living graphic which was originally published in 2016, but has been updated as required, most recently in 2019. This graphic, displaying the levels of automation in the automotive industry, is shown in Figure 1.4. The similarities to Sheridan's LOAs are apparent, however, these better capture the role of both the operator and the vehicle from a system level. Level 0 represents no automation, while Level 5 represents full autonomy. The area of interest for this research is Levels 1-3. In these levels the operator and automation work together with the goal of achieving enhanced performance. Levels 1 and 2 are typically considered operator aids, while Level 3 moves the operator to a supervise and intervene role. It is important to mention that these levels are not recommended for the aviation industry because control is suddenly handed back to the human when something goes wrong with minimal warning and/or minimal knowledge [23]. For aviation applications, the role of the operator and automation should be specified in the concept of operations (CONOPs) and not suddenly transition in an edge case [23]. This difference is important when defining the implementation of the automation system, but both methods highlight the need for differentiating the role of the operator. In lieu of an aviation scale, the SAE scale is utilized to discuss the relative role of the operator in a system.

While automation advances (i.e., SAE Driving Levels 1 through 3), the operator and

SAE J3016™ LEVELS OF DRIVING AUTOMATION

	SAE LEVEL 0	SAE LEVEL 1	SAE LEVEL 2	SAE LEVEL 3	SAE LEVEL 4	SAE LEVEL 5
What does the human in the driver's seat have to do?	You <u>are</u> driving whenever these driver support features are engaged – even if your feet are off the pedals and you are not steering		You are <u>not</u> driving when these automated driving features are engaged – even if you are seated in “the driver’s seat”			
	You must constantly supervise these support features; you must steer, brake or accelerate as needed to maintain safety		When the feature requests, you must drive			
What do these features do?	These features are limited to providing warnings and momentary assistance		These features can drive the vehicle under limited conditions and will not operate unless all required conditions are met			
	These features provide steering OR brake/acceleration support to the driver		These features provide steering AND brake/acceleration support to the driver			
Example Features	<ul style="list-style-type: none"> • automatic emergency braking • blind spot warning • lane departure warning 		<ul style="list-style-type: none"> • lane centering OR adaptive cruise control • lane centering AND adaptive cruise control at the same time • traffic jam chauffeur • local driverless taxi • pedals/steering wheel may or may not be installed 			
			<ul style="list-style-type: none"> • same as level 4, but feature can drive everywhere in all conditions 			

For a more complete description, please download a free copy of SAE J3016: https://www.sae.org/standards/content/J3016_201806/

Figure 1.4: SAE Standard for Levels of Driving Automation [89].

technology can be viewed as a team. *Ten Challenges for Making Automation a “Team Player” in Joint Human-Agent Activity* discusses the four basic requirements involved in joint activity among people: 1) an agreement to work together, 2) mutually predictable in actions, 3) mutually directable, and 4) maintain a common ground [59]. These requirements are similar to those identified by the Operational Considerations and the Human Role Pillar. When applying these requirements to automation integration, it results in ten challenges which are shown in Table 1.2. The focus of this work will be on Requirement 1 which is also highlighted as Challenge 1 for human-agent activities, the agreement to work together and its implications on the operator and system performance. It is given the term ‘Basic Compact’ and represents the agreement to work together based on some level of goal alignment. The Basic Compact and its limitations must be clearly understood by the operator, “breakdowns occur when a party abandons the team without clearly signaling his or her intentions to others” [59]. Challenge 9 highlights the other focus of this work, attention management. Team members must be able to direct each other to information of importance with enough time such that the information is relevant and without overwhelming the operator. The operator must have the time to understand and act upon the information received. An inability to overcome these challenges can lead to overall system failures.

The Basic Compact between the automation and the operator from a design standpoint is constant. However, the uncertainty and utilization of technology changes based on operator expertise and understanding. In this case uncertainty is the misunderstanding between what the operator expects the automation to be doing and what the automation is currently doing. A 2003 study by Naval Undersea Warfare Center (NUWC) looked at the interaction between automation and uncertainty, and discusses how automation both creates and mitigates uncertainty [56]. Expert operators are very aware of the uncertainty within automation, therefore they will wait for confirming evidence and not allow the technology full autonomy. Novice operators are less aware of the limitations in the automation, they

Table 1.2: Ten Challenges in Human-agent Teaming [59].

#	Challenge
1	A Basic Compact
2	Adequate Models
3	Predictability
4	Directability
5	Revealing Status and Intentions
6	Interpreting Signals
7	Goal Negotiation
8	Collaboration
9	Attention Management
10	Cost Control

exhibit higher levels of trust and do not search for individual confirmation [56]. Both of these details are shown in Figure 1.5. The study looked at how often operators of different skill levels checked automated data against “raw” data. The figure shows that the expert operators evaluated the reliability of the automation, however, novices and journeymen were less likely to check the data and relied on the automated information [56]. For an expert operator, he now must check the data as typically required by his job, and ensure proper functionality of the automation; while a novice may rely too heavily on an automated system such that he does not intervene when necessary. The need to double check automated information and differences between skill levels is important when looking at the operator’s workload and attention management.

Automation has the intent of decreasing workload, however, many times it just changes the form of the work, causing “the very systems designed to reduce the need for human operators [to] require more manpower to support them” [110]. This effect can be seen in the 2007 [13] and 2009 [12] studies, which combined the positions of a gunner and robotics operator to understand the effect on performance under differing levels of automation in a multi-tasking environment. The operator was in charge of local identification of targets for the gunner role and had tasks associated with the control of an Unmanned Ground Vehicle (UGV). Some of the results from the 2009 study are shown in Figure 1.6. The study

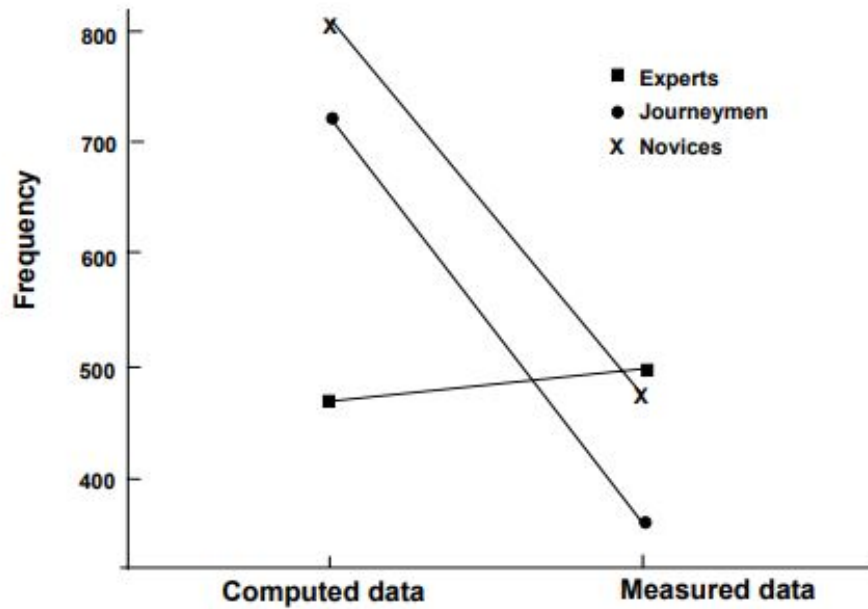


Figure 1.5: The frequency which operators of different skill levels used automated data and raw data in a NUWC study [56].

showed that the additional workload from the robot operator role increased the workload and decreased the operator's performance, however, it also showed that the increments in workload and decrements in performance are directly related to the amount of effort that the operator must contribute to controlling the UGV. Therefore, under the appropriate level of automation, one operator may be capable of performing both roles to a satisfactory level without requiring unattainable or unsustainable workload levels. When the appropriate level of automation is not assessed or automation's effect is not understood, the repercussions can be catastrophic; this will be considered the operator's cognitive uncertainty.

The catastrophes from not accounting for cognitive uncertainty in system design are becoming an increasing concern with further introduction of automation technology. An example of a system which failed to account for the operator's cognition is the Tesla Autopilot. This system utilizes a set of sensors to enable automated highway navigation; it is capable of auto-steering and adaptive speed control, enabling the system to automatically change lanes to optimize routes and avoid slower vehicles, however, Tesla has stated that "current Autopilot features require active driver supervision and do not make the vehicle

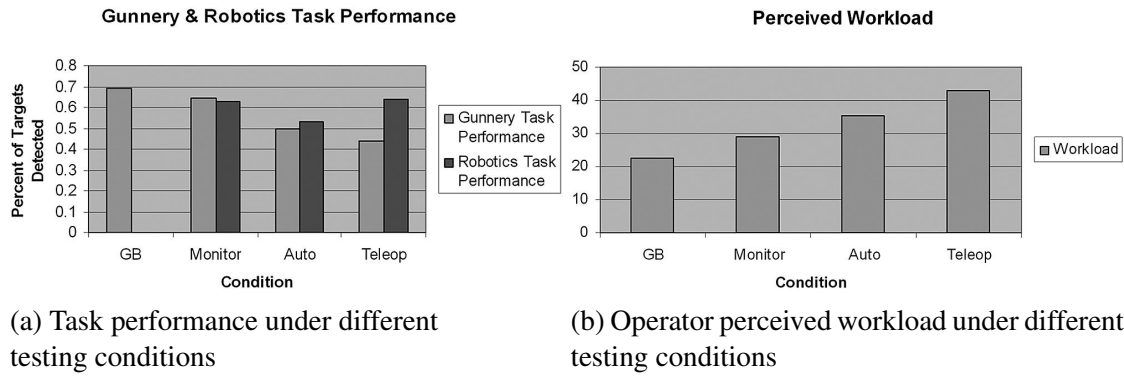


Figure 1.6: Results from a study assessing performance under different levels of autonomy for a gunner also controlling and Unmanned Ground Vehicle (UGV) [12]

autonomous” [109]. The stated functionality places the Tesla Autopilot in SAE Level 3 Automation, because the driving action must be conducted by the operator when the Autopilot is unsure how to proceed or the driver should intervene when the automation is about to take an inappropriate action. However, many operators of the Tesla Autopilot system would be considered novices. They do not understand the limitations of the automation and are entering into a Basic Compact without a full understanding. With low cognitive stimulus and no necessity to be a part of the driving action, novices rely solely on the automation and are not constantly double checking the automation with raw data. Due to this lack of involvement there are many stories about drivers falling asleep or reading while “driving”. A 2019 National Transportation Safety Board (NTSB) report discussed this issue after a Tesla crashed into a firetruck in 2018. The report called out the system design along with driver errors for primary causes of a highway crash; “the probable cause for the crash was the Tesla driver’s lack of response to the fire truck parked in his lane, due to his inattention and overreliance on the car’s advanced driver assistance system” [85]. Figure 1.7 shows that the operator was disengaged from the driving task for almost 20 minutes preceding the crash. The level of allowable inattention is being blamed directly on system design. These cognitive challenges in system design are not unique to Tesla, or the automotive industry.

Similar cognitive challenges and necessities to understand operator cognition are also appearing within the aerospace industry. The airlines industry continues to improve its

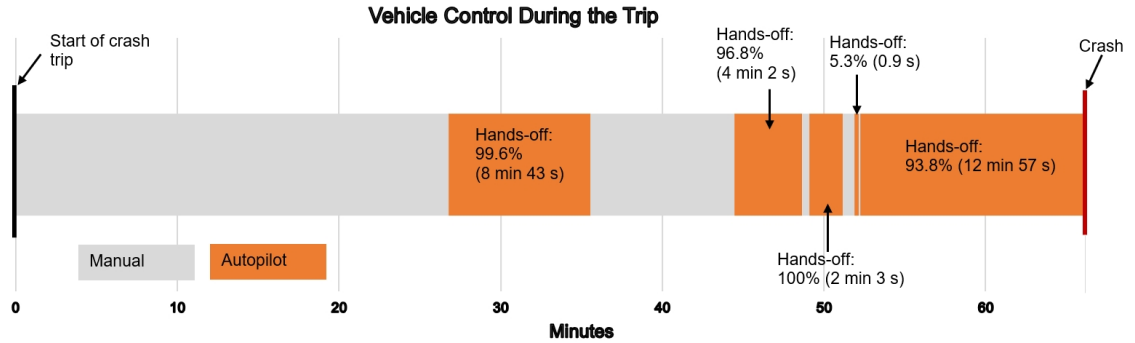


Figure 1.7: NTSB graph showing the amount of operator control before the 2018 Tesla crash [109].

safety record and remains the safest method of travel, aided by automation technology. However, the recent crashes of Boeing’s 737 Max highlight some of the challenges associated with increased automation and are of note because the automated system are switched on for 90% of a typical trip [117]. Although the crashes were blamed on design flaws and the plane’s Maneuvering Characteristics Augmentation System (MCAS), it also brought up concerns around pilot skills and intervention capabilities. A veteran pilot, John Cox, said about the incident that “‘automation dependence is not a cause, but it is a contributor’ to the disaster” [117]. Similar concerns were stated in Transportation Department Inspector General Calvin Scovel’s prepared remarks for the United States Senate subcommittee on Aviation and Space, “Boeing 737 MAX 8 accidents have suggested a possible link to one of the aircraft’s automation systems, raising concerns about pilots’ abilities to recognize and react to unexpected events” [97]. These incidents concerning the 737 Max have shown a greater need to understand pilot cognition and attention during the design process.

This need to further understand cognitive uncertainty during design has also been highlighted within the field of defense. The 2019 United States Air Force Science and Technology Strategy states that the “increasing complexity and speed of the battlespace means that the demands on combat decision-makers are outstripping the cognitive capacity of the unaided human” [96]. This statement captures the importance and trends towards automation integration. In the same report US Secretary of the Air Force Heather Wilson addresses

the importance of being able to capture the effect of this integration, “the advantage will go to those who create the best technologies and who integrate and field them in creative operational ways that provide military advantages” [96]. This becomes apparent when discussing technologies like the Boeing Airpower Teaming System, introduced earlier. When assessing this system of systems’ effectiveness it is important to understand the imposed cognitive load on the operator during different mission segments. As shown by the study on the gunner who shared the role of UGV operator, the level of cognitive load can be directly related to system performance.

As operator’s cognition levels in system design become important, it is critical that it can be assessed in the early stage of design to reduce uncertainty. Cognitive constraints drive many cost factors during its operational life and determine the system effectiveness during its utilization. The Yerkes-Dodson law, developed in 1908, is an early representation of the relationship between operator arousal and performance, but it did not grow in popularity until after the work by Hebb in the 1950s [108]. The Hebb/Yerks-Dodson hybrid is shown in Figure 1.8. It illustrates the bell-shaped relationship that performance has with operator arousal. The Tesla Autopilot shows performance degrade due to low arousal while the gunner experiment shows performance degrade due to over arousal. A lack of cognitive assessment during automation integration is causing an unsustainable amount of uncertainty in the operational assessment. The NTSB is blaming Tesla for design flaws and the military continues trending towards manned-unmanned teaming. With these trends, it is important that during the design process the operator’s cognitive load be captured (arousal) when introducing new technology such that an improved understanding on expected system performance may be made.

The system design process already conducts data capturing activities among many disciplines to properly create system requirements and assess a system’s functionality through trade studies. Raymer’s *Aircraft Design* highlights the different trades typically conducted during the design process, shown in Table 1.3 [87]. Design trades vary wing-geometry,

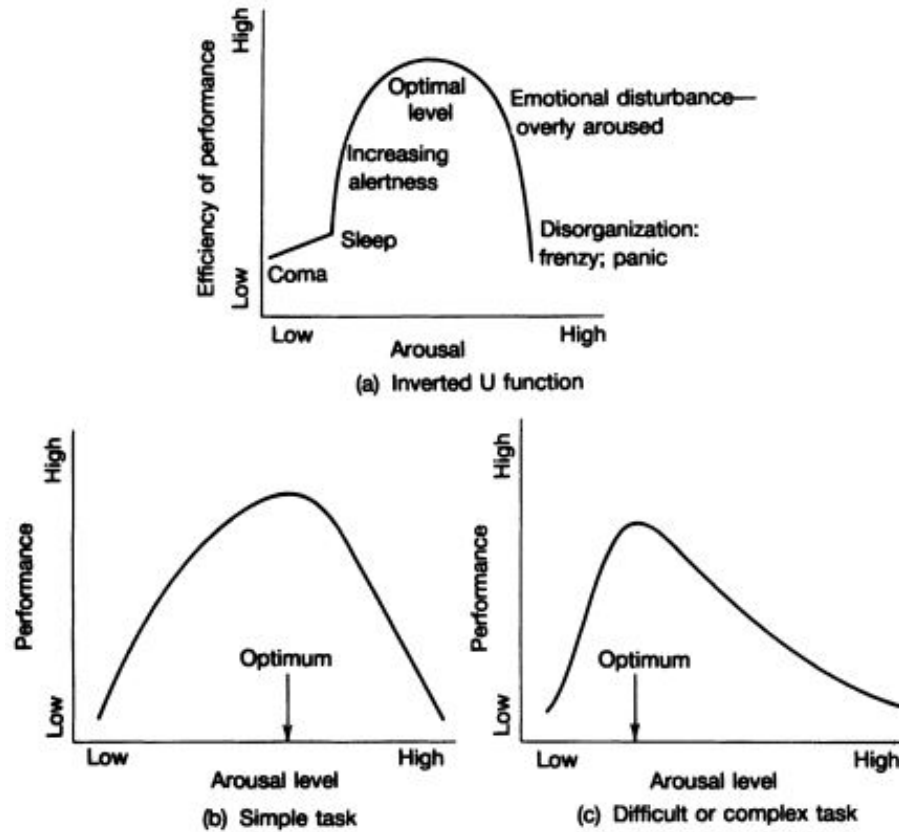


Figure 1.8: Hebb/Yerkes-Dodson Law relating arousal to performance for difficult and simple tasks [58].

propulsion, and configuration to satisfy a given set of mission and performance requirements, while requirement trades determine the sensitivity of the aircraft to design requirement changes [87]. The advancement in computing has facilitated a significant shift towards modeling and simulation (M&S), better enabling technology trades and reducing development times before proceeding to testing and evaluation [15]. These trades currently capture the physical characteristics of the system, but they do not convey any information about the operator, leaving high-levels of uncertainty around the operator’s workload and situational awareness. *Research Challenges in M&S for Engineering Complex Systems* discusses the lack of human aspects in M&S activities, “while there has been significant progress on understanding humans as decision makers, the utilization of this knowledge in M&S activities has been limited” [15]. The gunner and Tesla examples show how allowing

Table 1.3: Trades studies which are typically conducted during the aircraft design process [87].

Design Trades	Requirement Trades	Growth Sensitivities
T/W and W/S	Range/payload/passengers	Dead weight
A, Λ	Loiter Time	$C_{D,0}$ and K
t/c, λ	Speed	$C_{D,wave}$
Airfoil shape and camber	Turn-rate, P_s , n_{max}	$C_{L,max}$
High-lift devices	Runway length	Installed thrust and SFC
Fuselage fineness ratio	Time-to-climb	Fuel price
BPR, OPR, TIT, etc.	Signature level	
Materials	Design-to-cost	
Configuration		
tail type		
variable sweep		
number and type of engines		
maintainability features		
observables		
passenger arrangement		
Advanced Technologies		

this uncertainty to continue can lead to decreased system performance or outright system failure, therefore there is a desire to reduce the uncertainty around the operator's situational awareness and workload during early-design M&S activities.

CHAPTER 2

TECHNICAL APPROACH

This research is focused on reducing the operator-system uncertainty during performance evaluations for increasingly automated systems. This work will provide an increased understanding through in context analysis of operator workloads such that they could be utilized during technology evaluations. However, before jumping into operator workloads, it is important to provide some background by looking at the system development process and the resulting performance evaluations.

2.1 Background

The reduction of operator uncertainty requires an introduction and background of multiple fields. The first section discusses the development process including how system requirements are created and evaluated. It breaks down systems engineering concepts until the growth of model-based systems engineering (MBSE) can be discussed. MBSE is used in the design process to evaluate alternative designs based on system requirements, which often included emerging technologies. Second, operations research and analysis is introduced which is the field responsible for in context system analyses. Pairing these fields together enables the development of system requirements through operational evaluations.

2.1.1 Systems Engineering and System Development

Before discussing the field of systems engineering, it is important to define a system. This research will use The International Council on Systems Engineering (INCOSE) definition which states: “[a system is] a combination of interacting elements organized to achieve one or more stated purposes”[106]. This definition clarifies why it is important to include the operator in a system evaluation. The operator is a critical element to achieving the

stated purposed of any system that is not fully autonomous. Using this definition, INCOSE defines systems engineering as:

An interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem. Systems Engineering considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs. [106].

Systems engineering ensures that the system, when all of the pieces are brought together, fulfills both the system and thereby customer requirements throughout the duration of its life-cycle, or said simply “systems engineering is the practice of engineering from the systems viewpoint” [25]. The Vee model is a life cycle model that helps to visualize the systems engineering focus, with an emphasis on the early stages of the design process (concept and development stages)[106, 25]. The model is shown in Figure 2.1. It starts on the top left with user requirements and proceeds with steps of system decomposition and definition along the left side until it reaches the physical subsystem and system creation steps at the bottom, then it proceeds up the right side with integration and verification steps. It is critical to get the requirements definition and initial system concept correct with minimal amounts of uncertainty, because changes late in the design process or life-cycle are typically heavily constrained and costly. This concept is shown in Figure 2.2, depicting notional trends between current and desired/future design freedom, knowledge, and cost committed throughout the system development process [71]. As design freedom decreases and the cost committed increases, it becomes a challenge to change any system requirements. Therefore, bringing increased knowledge forward in the design process helps ensure better requirement definition and reduces re-work costs. One of the methods

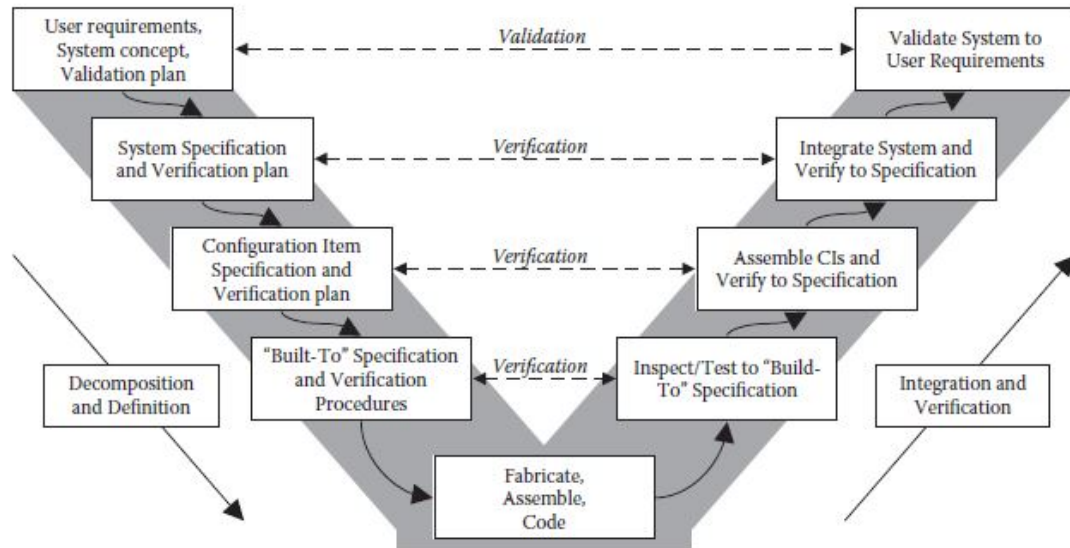


Figure 2.1: The Systems Engineering Vee Model [25]

of doing this is reducing uncertainty around system components, in this case the operator, which enables requirements to be created based on the new information.

Requirements

Requirements provide a set of goals that the system must attain to be considered a successful design. In the *Handbook of Systems Engineering and Management* chapter “Discovering System Requirements”, it formally defines a requirement as: “a statement that identifies a capability or function that is needed by a system in order to satisfy its customer’s needs” [6]. In order to improve the system design process when automation technology is integrated, designer’s need the capability to specify requirements based on expected operator performance. “Discovering System Requirements” has a number of criteria which must be satisfied in order to define a good system requirement [6]. These criteria are outlined in Table 2.1. This provides a very comprehensive list of requirements qualities, including what properties the requirement needs along with how it should be written to avoid any complications either internally or externally. A simpler list for system requirement criteria can be found in the *Department of Defense Handbook*, which are detailed in Table 2.2 [22].

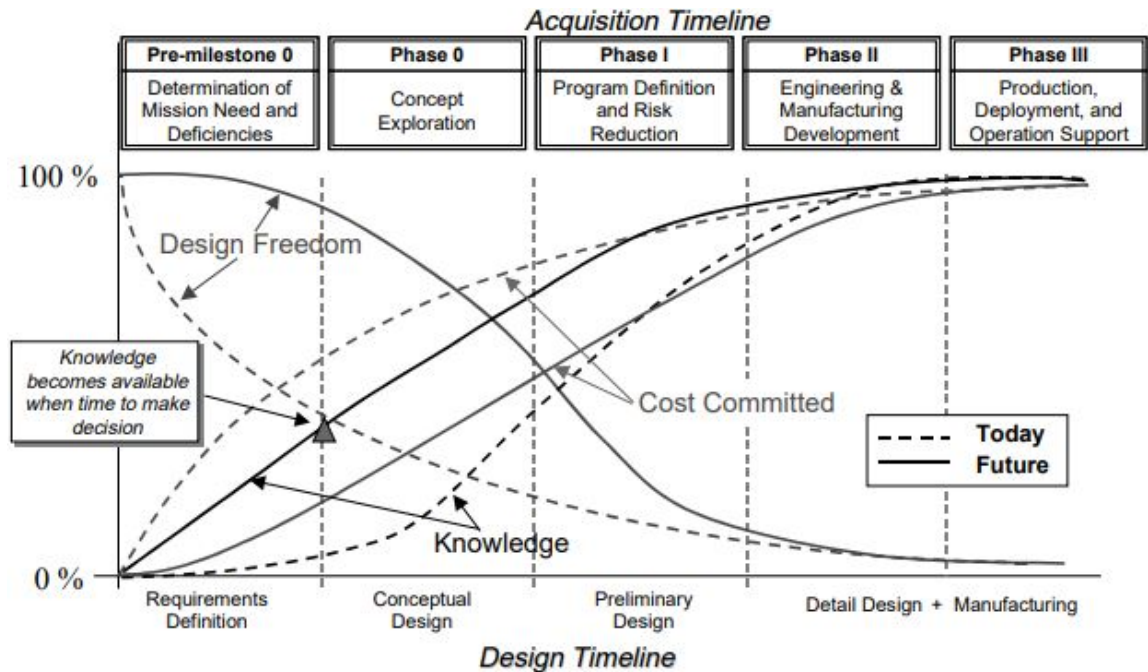


Figure 2.2: The relationship of design freedom, knowledge, and cost committed throughout the acquisition timeline [71]

There are many similarities between the two sets of criteria, but since this work is focused on aerospace and defense, it will focus on the Department of Defense (DoD) criteria for a requirement. These 7 traits are critical when adding additional system requirements, therefore when reducing uncertainty it is important that the methodology allows these criteria to be satisfied. This section focused on what requirements are, however, for reducing cognitive uncertainty it is also important to look at how requirements are defined and evaluated throughout the design process.

Requirements Analysis

Requirements analysis makes sure that the requirement set for the system achieves the designed outcomes. From the DoD Handbook, requirements analysis has four primary purposes:

- 1) Develop warfighter capabilities and objectives;
- 2) Define initial performance capabilities and objectives and refine them into

Table 2.1: Qualities of a Good Requirement according to the *Handbook of Systems Engineering and Management* [6].

#	Quality	#	Quality
1	Describes What, Not How	15	Verifiable
2	Atomic (or Unitary or Single-Minded)	16	States Its Units of Measurement
3	Allocation	17	Identifies Applicable States
4	Unique	18	States Assumptions
5	Documented and Accessible	19	Usage of Shall, Should, and Will
6	Identifies Its Owner	20	Avoids Certain Words
7	Identifies Its Target	21	Might Vary in Level of Detail
8	Approved	22	Contains Date of Approval
9	Traceable	23	States Its Rationale
10	Necessary	24	Respects the Media
11	Complete	25	Distinguishes Number
12	Is Not Written Negatively	26	Consistent
13	Unambiguous	27	May Use Parameters
14	Is Not Always Written		

Table 2.2: Characteristics of Good Requirements according to the *Department of Defense Handbook* [22].

#	Characteristic	Description
1	Achievable	The desired threshold must be technically realistic and an affordable cost
2	Verifiable	Uses clear language to define expected performance and functional utility in a manner allowing verification of the objective, preferably quantitatively
3	Unambiguous	Only one possible meaning which can be uniquely tested and verified
4	Complete	Contains all of the information necessary to interpret and verify
5	Written in performance terms	Expressed in terms of need not solution, focuses on “why” and “what”
6	Consistent	Agrees with all other system requirements
7	Appropriate detail	Contains an acceptable amount of detail for the design level as to avoid unintentionally constraining other levels

acquisition requirements;

3) Identify and define constraints that limit the solutions; and

4) Define functional and performance requirements based on warfighter provided MOEs [measures of effectiveness]/MOSs [measures of sustainability]/MOPs [measures of performance] [22].

This work primarily looks at the third purpose, identification of constraints by reducing the uncertainty around the effects of automation integration. By reducing the uncertainty, it will result in a better requirements analysis. Requirements analysis should provide a clear understanding of the system's functions, performance, interfaces, and any other requirements and constraints [22]. The functions focus on what the system has to do, but performance is interested in how well the functions and resultant system have to perform [22]. Going back to the experiment involving the gunner/UGV operator and his degraded performance in the dual-role, it is important that requirements analysis can define an acceptable level of degrade and anticipate the performance of this system after a level of automation is added. Proper requirements definition and analysis are essential steps within the system development process.

Development Process

Design requirements guide future steps of the development process, therefore it is important to understand how the process works and how the requirements are utilized throughout. The stages of the development process were shown in Figure 2.2 and are discussed in the work by Mavris and DeLaurentis [71]. Their work is a more general adaptation showing the ideal future trends for the DoD acquisition process. The DoD process is shown in Figure 2.3 and discussed in *Pre-Milestone A and Early-Phase Systems Engineering: A Retrospective Review and Benefits for Future Air Force Systems Acquisition* [83]. The development process can typically be thought of as having five unique phases. The phase of most importance to this work is the “concept refinement” phase, which is focused on

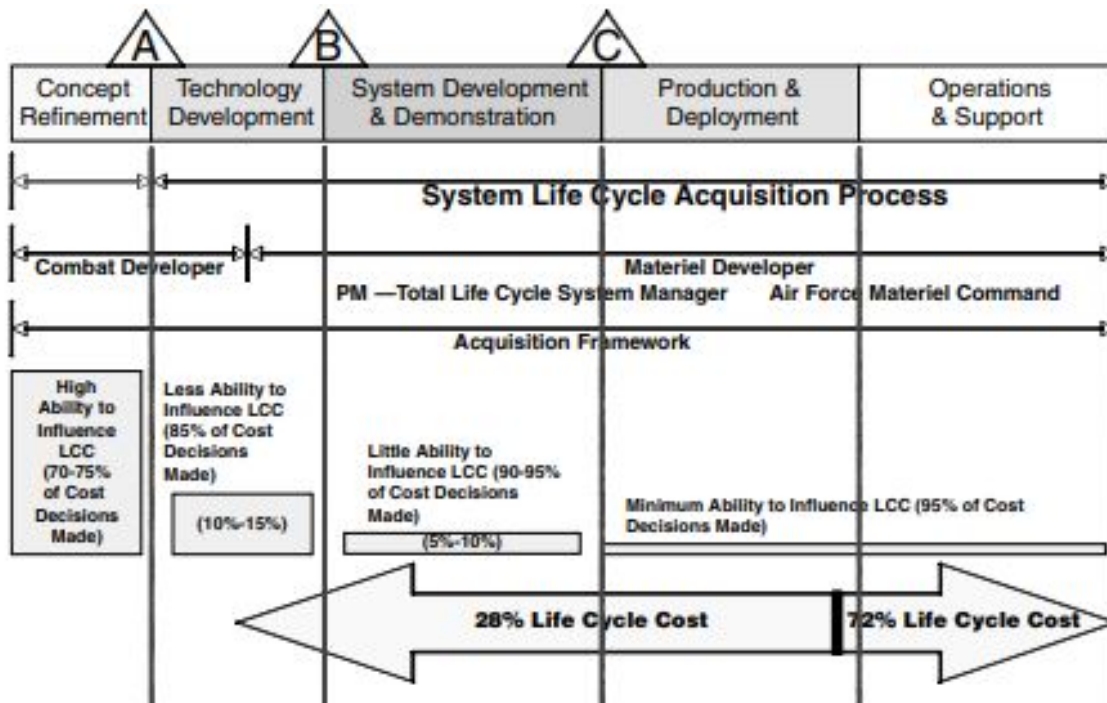


Figure 2.3: Department of Defense life cycle acquisition process with reference to costs committed and program expenditures [83]

the requirements definitions. This phase is responsible for most of the cost decisions for the entire project, therefore plays a large role in the success of the system. The system requirements define the concept space which will be utilized during the exploration phase, and helps specify the criteria by which systems and design decisions will be evaluated.

The steps that should be conducted early in the design process are outlined in the work by Schrage, DeLaurentis, & Taggart using the Concept Development and Systems Engineering (CDSE) process [95]. Their work discusses the importance of making the appropriate design trades in the early phases of the design process, and introduces a methodology that is intended to improve “the execution of systems engineering in the earliest stages of the acquisition/development process for complex systems” [95]. CDSE includes seven major activities and the associated tools which should be used to guide concept development; the process is shown in Figure 2.4. The first step in this process is utilizing Quality Function Deployments (QFDs) to translate top-level requirements into system capabilities. If the current system is not capable of attaining all desirable criteria then technology and

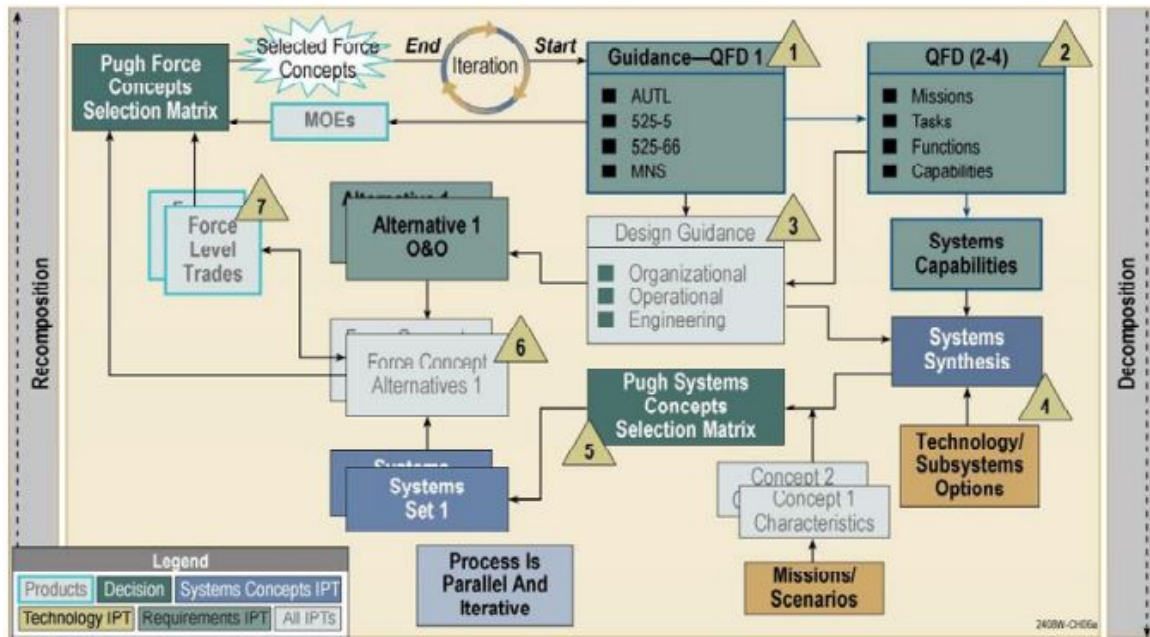


Figure 2.4: The Concept Development Systems Engineering (CDSE) Process [95]

subsystem options are explored. Once some system concepts are synthesized, they are tested against expected scenarios to assess their performance. Utilizing the performance assessments and a Pugh Selection Matrix, the decision maker can select the desirable system characteristics. Traditionally system capabilities could be thought of as performance characteristics, however, as automation is added it is important that operator characteristics be better modeled and captured for each concept during the mission assessment.

The CDSE process is based on Schrage's earlier work on Integrated Product/Process Development (IPPD) [95, 94]. Since so many costs are committed early and design requirements restrict later parts of the development process, it is critical that many alternatives be evaluated early in the process, before the design is specified. The IPPD Methodology is shown in Figure 2.5 with the center column describing the steps in top-down design. The seminal paper, written in 1999, applies the methodology to increase affordability during technology integration on rotorcraft [94]. The steps in IPPD are relatable to the early development phases discussed above and should be conducted before a design is manufactured. Generating and evaluating alternatives attempts to ensure that the correct design is

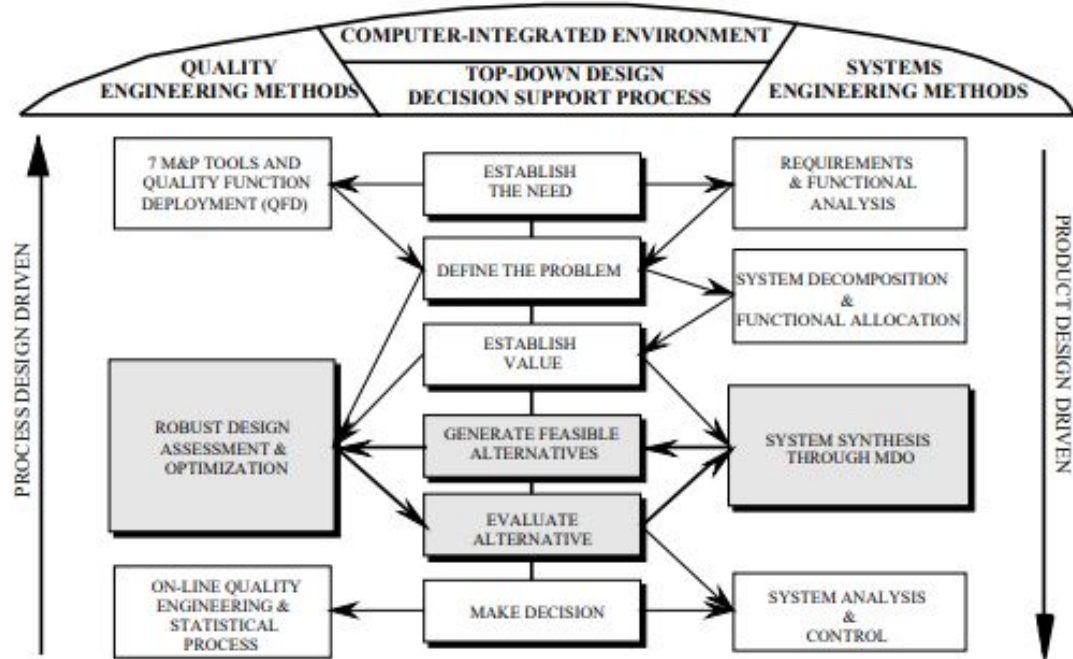


Figure 2.5: Integrated Product/Process Development (IPPD) [94]

chosen based on the system requirements, not just a satisficing design. A common method of evaluating alternatives is through models.

Model Based Systems Engineering

Models are undergoing a continually growing importance within engineering disciplines. They enable quicker and cheaper evaluations for a multitude of systems, before proceeding to scaled or full-size prototypes. This utilization has spawned a field of engineering which focuses on the effective utilization of models within systems engineering, Model Based Systems Engineering (MBSE). Dickerson & Mavris provide a brief overview of the past five decades of growth in MBSE [24]. This overview brings up Stephen Hawking's attributes of a good physics model: 1) simple, 2) mathematically correct, 3) experimentally verifiable; and it relates these attributes to desirable modeling techniques within MBSE [24, 48]. Modeling methods provide new analysis techniques and data throughout the system development process. Through modeling, many design requirements can be assessed and

amended before a final concept is chosen. This is seen in work like Lewe et al. which uses agent-based modeling (ABM) to identify design requirements for a personal vehicle system (PVS) [63]. The model assessed features such as number of passengers and cruising speed to determine expected market share for the PVS. Their work focused on requirements exploration, prior to the design stages, to better understand the impacts before feeding these requirements into technology integration. A similar approach should be utilized when trying to understand desirable automation integration on systems. This approach checks the impact of changing the operator's awareness and workload, prior to spending research dollars on figuring out how the technology itself can be created.

Technology Selection

Technology selection is often an important part of the development process. When there is not a suitable design within the concept space (constrained by the system requirements), technology must be developed and integrated to open up the concept space. The process for technology integration has been formalized for systems by Mavris & Kirby [72] and later at the system of systems (SoS) levels by Mavris & Sudol [73]. Figure 2.6 shows the Technology Identification, Evaluation and Selection (TIES) methodology for both systems and SoS. Both categories of design problem with technology integration go through a similar process, however, SoS must assess ways (tactics) along with technologies to understand the holistic impact [73]. The TIES methodology is similar to the IPPD process discussed above but with increased definition around the steps involving technology. TIES starts with problem definition and the specification of system requirements, then based on those requirements defines an applicable concept space (specifying the design features that will be considered throughout the concept exploration). Based on the concept space, an initial concept is chosen to be the baseline. This baseline provides a benchmark to compare other concepts against and ensure model accuracy; it usually represents a well understood, existing system. Following the specification of a base line, alternative concepts are chosen

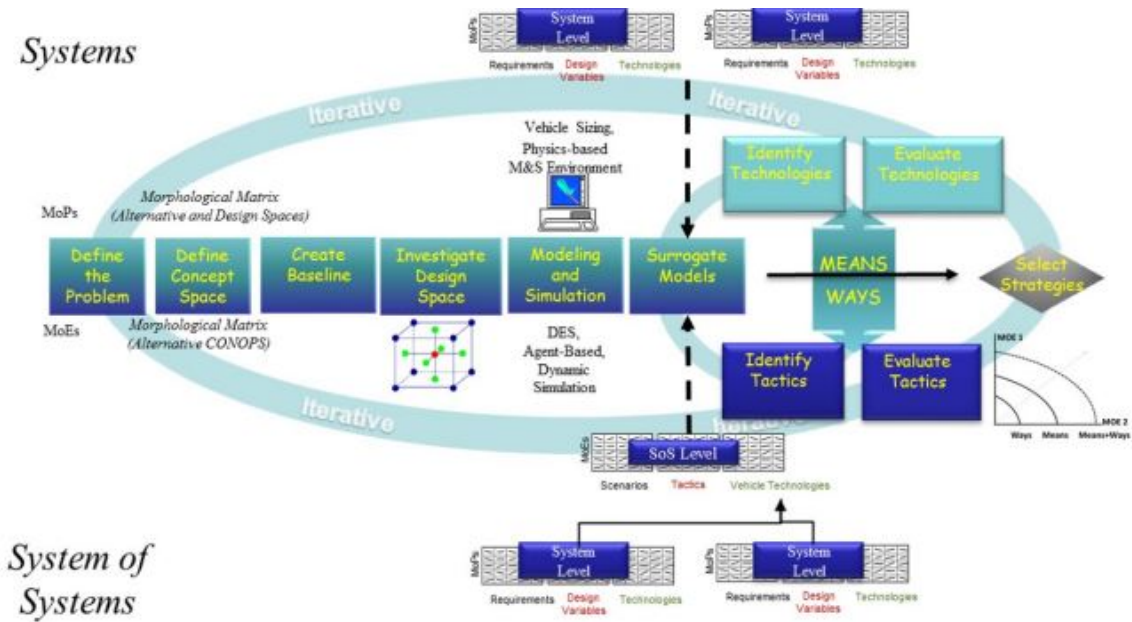


Figure 2.6: Technology Identification, Evaluation, and Selection (TIES) for Systems (just using the top path) and System of Systems (the top and bottom with iteration) [73]

from the design space, typically based on a Design of Experiments (DoE) which maximizes experimental effectiveness. These concepts then must be assessed such that they can be compared.

TIES uses physics-based modeling to capture the effects of technology integration within each concept. It captures the technology effects through the inclusion of k-factors, which are scale factors on discipline specific factors, to create surrogate models which can be used during technology trade studies [72]. In example, a model for assessing new engines on an aircraft may capture the effects such as range and cost for different fuel burn rates and system weights on a mission set, then based on the expected changes to fuel burn and weight a technology will be chosen. This method has been applied to UAV propulsion techniques [100], aircraft systems [70], and launch vehicles [105]. TIES is capable of helping with the concept evaluation if appropriate discipline specific metrics can be measured during the mission modeling phase.

2.1.2 Operations Research

As introduced by the TIES methodology, the system's design process requires concept modeling to understand the impact of technology integration. When looking at systems which integrate automation technology, it is critical that these models capture the operational environment in which the system is expected to operate. The field that specializes in the functional assessments of systems is Operations Research (OR). *Methods of Operations Research*, originally written in 1946 as a classified document, is considered one of the fundamental works and opens with the following definition for OR: "a scientific method of providing executive departments with a quantitative basis for decisions regarding operations under their control" [80]. There are some early references to OR, but it really started to emerge after World War II. The British Military was the first to combine the weapons/equipment performance evaluation with the analysis of the operations to determine the interaction with tactics, and to what degree tactics dictated the weapons selected [99]. Some of ORs first major successes appeared in tactical changes during the Battle of the Atlantic: 1) based on analysis, depth charge settings were changed which increased efficiency somewhere between 400 and 700 percent, and 2) convoy sizes were increased after studies showed that merchant ship loss rate decreased while holding escorts constant, reducing the number of overall crossings by one-third and U-boat attacks by almost an equivalent amount [99]. OR was thought to be different from other branches of engineering because it focuses on the use of the product so it reports to the user, as opposed to other disciplines of engineering which are involved in the construction of a system therefore report to the builder or manufacturer [80]. OR is focused on providing quantitative insights about possibly non-quantitative aspects to better inform a decision maker.

Since World War II, OR has continually grown in prevalence in both military and industry. Studies have moved earlier in the development process such that they are currently being used to drive acquisitions and system requirements. The 87th Military Operations Research Society (MORS) Symposium presented a collection of recent OR studies, such

as a force structure analysis for the United States Air Force - Europe [37] and “Modeling the Interplay Between Capabilities and Tactics to Reach Enhanced System of Systems Performance” [40]. The growth in OR is expected to continue, as in 2016 it was added as a priority in the defense acquisition and requirements review process: “The Secretary of Defense shall ensure that analytical organizations within the Department of Defense, such as the Office of Cost Assessment and Program Evaluation, provide resources and expertise in operations research, systems analysis, and cost estimation to the Joint Requirements Oversight Council to assist the Council in performing the mission in subsection (b)” [113]. Therefore, it is important that systems with high levels of automation technology, such as the Boeing Airpower Teaming System, can be properly captured within an OR analysis.

Operations Research Steps

Operations Research problems are broken down using a multi-step process. The process itself is considered between five and seven steps, with some sources combining steps, but the overall approach remains consistent. Some different discussions about the process are contained in *History of Operations Research in the United States Army, Volume I: 1942-1962* [99], *Maynard’s Industrial Engineering Handbook* [121], *Quantitative Techniques for Managers* [74], and Castello’s “The Operations Research Problem Solving Process” [10]. This overview will utilize the 7-step process from *Maynard’s Industrial Engineering Handbook* and the three-phases discussed by Castello. The OR process is shown in Figure 2.7 with the phase breakdown shown on the right side. Regardless of the number of steps, the process fits well into three phases: formulation, analysis, and interpretation. During the formulation phase, the analyst breaks down the problem and creates a research plan; the analysis phase is focused on carrying out the research; then the interpretation phase is about the research findings. The steps provide further detail inside each phase, but each phase must be conducted to achieve an OR study.

The OR process starts with a formulation phase, where the analyst orients himself to

the space, defines the problem, and collects relevant data. Since OR is reliant on the use of a system, it is important that the analyst spends time understanding the space prior to defining the problem and deciding which metrics will be captured. Once a sufficient understanding of the space is gleaned, the analyst can proceed to problem definition. This is often considered the most difficult OR step, because it defines the scope, necessary assumptions, and the desired results [121]. A clear problem definition contains three broad components: 1) unambiguous objective, 2) specification of factors which affect the objective, and 3) specification of constraints on course of actions [121]. A good problem definition provides the detail for all future steps of the process. It states in words what will later be captured in the model so it must be thoroughly detailed. The last step in formulation is data collection. Data is typically collected through observing the system in action or obtaining relevant standards, but can also be captured for a specific system through surveys or questionnaires [121]. The quality and availability of data has a direct effect on the problem definition (it must be re-stated if data cannot be procured) and overall model formulation.

With problem formulation complete, the analyst moves into the analysis phase. The MBSE overview section above introduced modeling from a SE perspective, but that focus is often on the system; OR is focused on the system and environment interaction. A model is defined here as “a selective abstraction of reality” [121]. Using the problem definition, the analyst attempts to create a model at the appropriate fidelity, such that it captures all key elements of the systems but is not overly detailed. There are multiple modeling levels and techniques that can be used and are covered in the next section. Once a model has been created, a solution can be found. The form of the solution is dependent on the expected results from the problem definition. Based on the desires of the problem, an optimal solution can be found, however, this is often a time consuming process so more focus is on a satisficing solution which ensures some minimal set of criteria are met [121]. Once a solution or set of solutions has been found, the solution must be verified and validated. The analyst must make sure that the solution intuitively makes sense. The solution in OR is often found to

be infeasible, which requires revisiting the model formulation and its assumptions [121]. Once a solution is deemed sensible, the analyst should check the robustness of the solution. Slight variations to the system and environment should be made within the model to understand the implications. The output is only as good as the model, and since the model is a selective abstraction it is important to understand the implications of those bounds. This output analysis helps ensure that the decision maker better understands the concept space.

The last phase of the OR process is the interpretation phase. As discussed previously with the definition of OR, this field is not meant to make the decision, merely provide quantitative analysis to decision makers. Therefore, once a solution is found and presented, if a decision maker decides to proceed then the last step in the OR process is implementation and monitoring. This stage ensures the accuracy of the models, and looks for early signs of variance between the study and the implementation. An OR study is only valid for as long as the assumptions hold, therefore if there is a drastic change in the environment then the study must be re-conducted. This OR process has been utilized in almost every industry including: gas production, commercial aviation, manufacturing, and services [121]. As systems become increasingly dependent on implementation factors, the best solution is dependent on how it will be utilized. It is important that an OR study be carried out early in the development process to capture operational characteristics and facilitate technology trades.

Types of Modeling

In the analysis phase, an analyst must decide which type of model is appropriate based on the current problem definition. There are four broad classes of models (in order of abstraction, least to most): physical models, analogic models, computer simulation models, mathematical models [121]. Physical models are actual, scaled-down versions of the desired system. Although prevalent in aerospace, such as wind tunnel models, they are uncommon in OR due to the challenges associated with capturing large, complex systems,

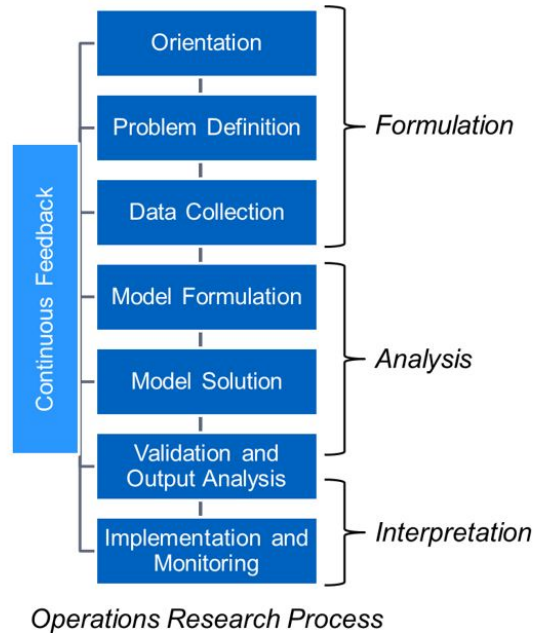


Figure 2.7: The 7-step Operations Research process showing the typical three-phase breakdown. Adapted from: [121, 10]

which are the common focus of OR studies. The second class of model is analogic models. This class uses a physical analog to describe the system instead of an exact scaled-down system. The most famous example used an anthill on a raised platform and little mounds of sugar on separate platforms to represent a military depot and various demand points, analyzing the ants paths to determine optimal networks, to demonstrate that valid OR analysis was possible without resorting to computer models [121]. Although this is an interesting class and demonstrates that OR does not always need mathematical analysis, this class is difficult to integrate with models capturing the other disciplines involved in the development process. Another class of models are mathematical, which are considered the highest level of abstraction. These models capture the critical elements of the systems in either a probabilistic or deterministic manner using three main elements: decision variables, constraints, and objective functions(s) [121]. These models have traditionally been the most commonly identified within OR, and work great when a representation can be determined. However, many systems are difficult to translate all the way to the mathematical model, which brings up the last class of models, computer simulation.

Computer simulation models are the focus of this work, because in systems development it can be challenging to abstract all relevant information to a mathematical model. Also creating a physical or analogic model does not enable rapid trade space exploration because they are costly and time consuming to produce. This class of models extracts system properties into a computer program to simulate the behaviors. Driven by advances in computational power, simulation models continue to grow in popularity with “one highly desirable feature: they can be used to model very complex systems without the need to make too many simplifying assumptions and without the need to sacrifice detail” [121]. However, computational limits and time required to develop the model still factor into the model formulation. The Air Force Research Laboratory (AFRL) has broken computer modeling and simulation activities into four levels: campaign, mission, engagement, and engineering/physics, which are shown in Figure 2.8 [8]. Each level is focused on a different scope in an effort to provide insights appropriate to the problem definition.

The campaign level is focused on providing strategic insights, such as the force structure analysis for the United States Air Force - Europe [37]. Modeling and simulation activities at this level are interested in understanding the entire battle space and its evolution over time, so it capture movements, losses, logistics and resupply rates. The closest analogy in business may be expansion into a new market and modeling market penetration over time. This level of computer simulation is too aggregated to assess system technology. Reducing the modeling aggregation, mission level and engagement level computer models often show some overlap. The mission level is still focused on the system of systems (SoS), however, the time scale is shorter. It allows for SoS course of actions to be modeled and better understand, but typically does not output results at the individual system level. The work by Harris et al. represents a mission level model that assesses UAVs swarms for the monitoring and detection of migrants in the Mediterranean [43]. System level metrics are still specified as inputs, however, modeling is aggregated and output metrics look at the effectiveness of the SoS. Engagement level continues to decrease the level of aggre-

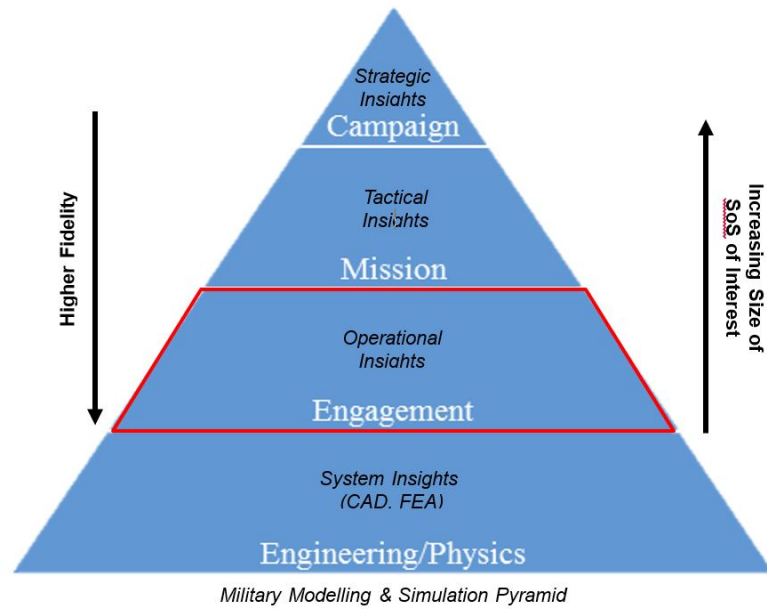


Figure 2.8: Levels of military modeling and simulation activities. Adapted from: [8]

gation and focus on a particular system, although many engagement studies include few versus few agents of interest. In example, Harper’s work shows the effect of aircraft design parameters (wing area, taper, engine scale) on a strike aircraft’s probability of arrival and survival [42]. The engagement level provides a good balance between detailed system modeling while still understanding its interaction with the environment. The final level of modeling activity defined by AFRL is the engineering/physics level. This level includes detailed models such as finite element analysis (FEA) and computer-aided design (CAD). However, it falls outside ORs domain. Sometimes detailed physics models are used in the other levels but physics models themselves do not fall within the OR definition provided above. Engagement type models are of the greatest relevance to this work due to their system-level focus.

Related to model level, modeling approaches must also be discussed. Inside computer simulation models there are three categories of modeling approaches: system dynamics, discrete event simulation (DES), and agent-based modeling (ABM). The variations in complexity and resolution of each modeling type is shown in Figure 2.9. System dynamics

models capture the behavior of nonlinear, highly interconnected systems over time [112]. They use feedback loops and flows in the form of differential equations to describe systems using a top-down approach. System dynamics models are simple to set up if all the information is available and useful for higher level modeling (like campaign), but are unable to handle stochastic or continuous problems therefore are not suitable to system-level technology evaluations [49]. DES captures systems with state changes based on events, the state change occurs instantaneously upon the events occurrence [112]. DES models are able to be built top-down or bottom-up, but are typically built top-down and require event, time, and state changes to be defined precisely [49]. ABM is the most recent and rapidly growing field. It has the highest resolution and complexity of the three approaches; ABM is capable of being implemented to solve any problem, however, it is also more cumbersome so should only be employed where a system dynamics model or DES is not appropriate [49]. ABM details each system through a bottom-up approach then captures the overall characteristics of the independently defined agents. Each agent contains a rule-set through which it is capable of dynamically interacting with changes in its environment; “agents perceive and act within their situated environment to reach their goals” [112].

To reduce the uncertainty around automation technology, OR studies must be conducted early in the development process that are capable of capturing the effect of integration on the system’s operational effectiveness. This is similar to Harper’s use of a computer simulation model at the engagement level to conduct an OR study that allowed mission effectiveness to be used as the objective in vehicle design [42]. However, an automation focused model must capture changes in the operator’s workload, situational awareness, and system performance. This type of model will facilitate automation technology trades based on system effectiveness. By capturing the effectiveness of systems with differing levels of automation in an engagement level model during the initial requirements analysis, the mission effectiveness can be assessed before proceeding down the development process.

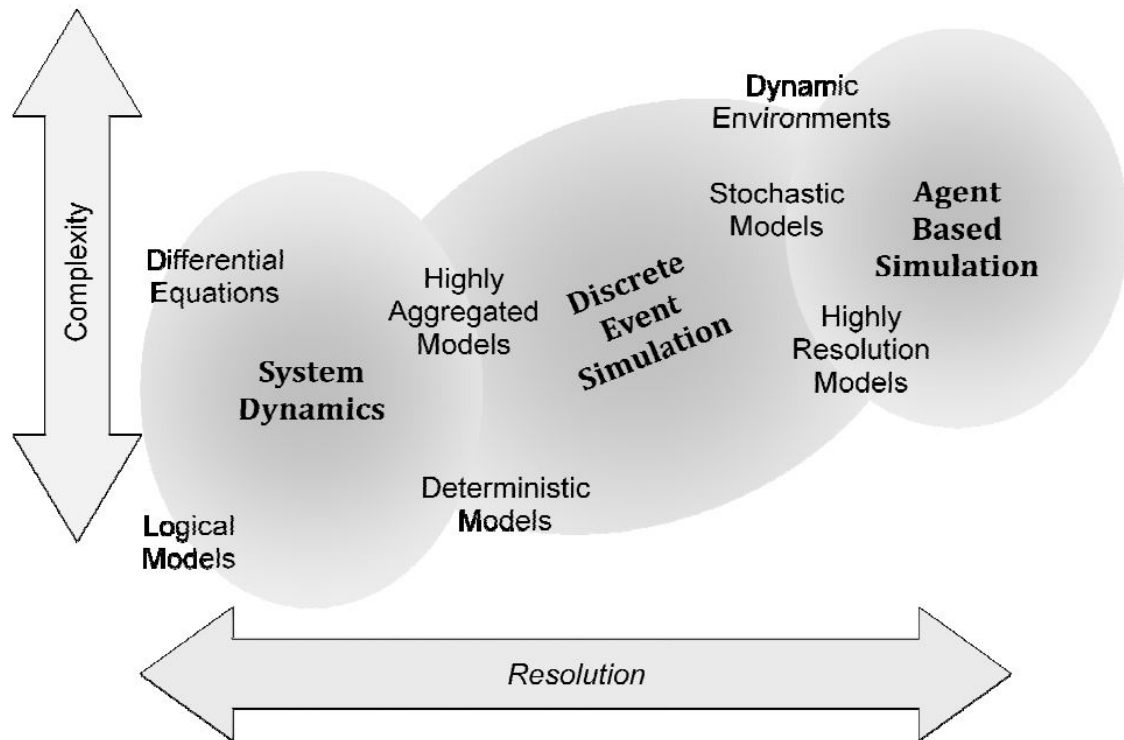


Figure 2.9: Modeling and Simulation Spectrum for Engineering [49]

Modeling Behaviors

It is important to look at how current systems are modeled within OR studies to understand the challenges with capturing automation technology integration. At this point, ABM will be the focus of discussion, because “while all M&S paradigms can provide some support to autonomous systems, the agent-based simulation paradigm is of particular interest, as autonomous systems reflect characteristics similar to those of software agents” [112]. Although the current focus is on lower levels of automation the same correlation between system characteristics and software agents exists. Figure 2.10 shows the general taxonomy of an intelligent software agent. A software agent is composed of three external domains (perception, action, and communication) and four internal domains (sense-making, decision making, memory, and adaptation) [112]. Starting with external, the perception is how the agent understands the environment around itself. For autonomous systems, this represents sensors which are used to interpret its surroundings: radars, cameras, light sensors,

etc. As the agent traverses through the model these perceptions will change as dictated by the virtual environment. The communication represents its capability to exchange information with other agents in the simulation. Lastly, the actions represents abilities the system can carry out when tasked by its internal domains, typically developed through some functional decomposition of the system [112].

Transitioning to the internal domains, sense-making receives the inputs from communications and perception then maps that information appropriately to the memory domain. This domain is responsible for data correlation, data fusion, and coping with incomplete or contradictory data. The memory domain stores all of the agent's information, while adaptation updates information in memory based on current goals, tasks, and desires. Decision-making then uses the information currently in memory to perform reactive and deliberative methods, triggering appropriate actions and communications [112]. The internal domains are typically harder to capture because these are the intangibles of an operator based system. The internal domains are an area of continuous development, but two approaches stand out in current OR: finite state machines and behavior trees.

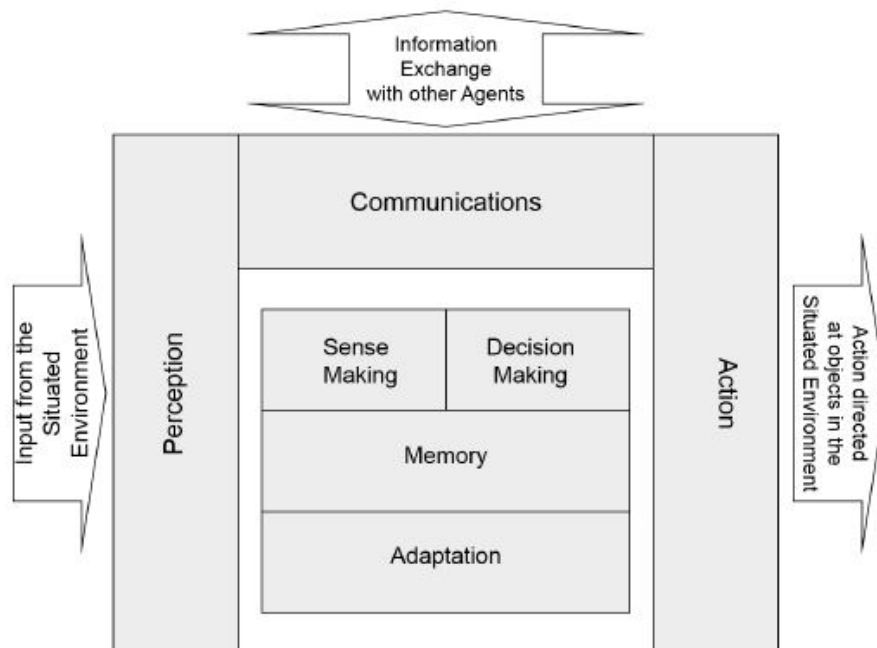


Figure 2.10: General Taxonomy of an Intelligent Software Agent [112]

The majority of research into modeling the internal domains was created for video games. As developers searched for increasingly sophisticated artificial intelligence for non-player characters, modeling techniques were improved and new techniques were created. Finite state machines (FSMs) dominated the control and decision making of automated players until the mid-2000s [118]. A FSM has three primary components: states, transitions, and actions. A simple, high-level representation of an FSM that would control Ms Pac-Man is shown in Figure 2.11. The states are the circles in the diagram. They store information about the task currently being performed and hold the actions to conduct while in that state (not shown in the figure). These actions create tasks for the external domains of action and communication. While in a state, the transitions are monitored to determine when to change states. In Figure 2.11, the transitions are shown as the connecting arrows with the conditional required for a state transition written next to the line. FSMs do not keep track of multiple states, they remain in one state and execute the actions in that state at some specified interval until a transition is activated. FSMs fall from popularity occurred because of two significant shortcomings. Since each state must be explicitly defined along with its actions and transitions, FSMs can rapidly increase in complexity (the number of possible transitions is the number of states squared) and therefore are computationally limited when looking at sophisticated systems [118]. The expansive number of transitions was attempted to be overcome using hierarchical finite state machines, however, “reusing transitions isn’t trivial to achieve, and requires a lot of thought when you have to create logic for many different contexts” [11]. The second issue with standard FSMs is that they are not dynamic or flexible. Once tested and debugged, FSMs have limited ability to adapt or evolve, resulting in very predictable, hard-coded behaviors [118]. In order to overcome these challenges, a new method for internal domain modeling was developed.

Behavior trees (BTs) were developed to overcome the complexity and predictability shortcomings seen in FSMs. The seminal work discussing BTs to control intelligent agents was presented at the 2005 Game Developers Conference titled “Handling Complexity in the

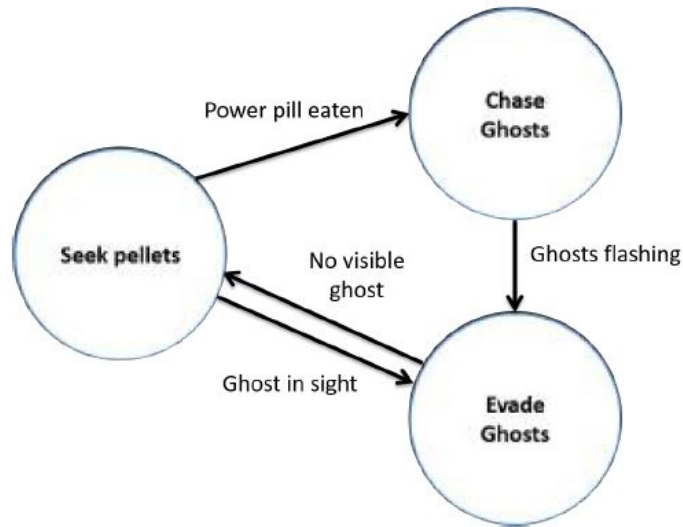


Figure 2.11: A simplified, high level representation showing the primary components of a finite state machine for controlling Ms Pac-Man [118]

Halo 2 AI” [53], and since then BTs have been heavily utilized to create more intelligent AI. A sample behavior tree for Ms Pac-Man seeking pellets is shown in Figure 2.12. Unlike FSMs which stay in their last state until a transition occurs, behavior trees execute from the top each time a stimulus event occurs. BTs are composed of three node types: sequencers, selectors, and decorators. A sequencer is shown as the blue rectangle at the top of the tree. This block first evaluates the success of the child node on the left (move). If that child node succeeds, then the next child node of the sequencer is evaluated until all children have been evaluated. If the last node is successful thus all children nodes were successful, then the sequencer succeeds, otherwise it fails. Continuing the node type discussion, selector nodes choose between children nodes. A selector node is shown as the red rectangle to decide how Ms Pac-Man should move. There are two types of selector nodes, priority and probability. Probability selector nodes choose the child node based on designer set probabilities, enabling some stochastic behavior. If the chosen child node fails, then the selector fails. However, for priority selectors the child nodes are in an ordered list. The first child node is evaluated, if it succeeds then the selector succeeds, but if it fails then the next child node is chosen. A priority selector fails if none of the child nodes succeed. The

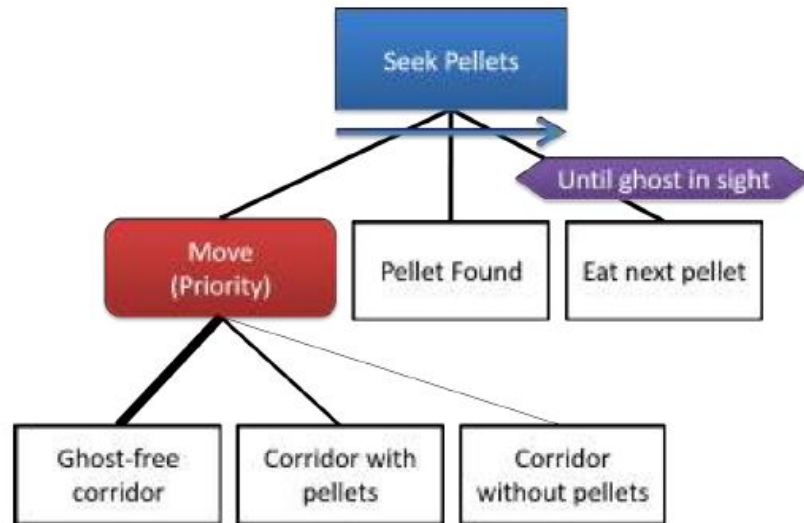


Figure 2.12: A simplified behavior tree for controlling Ms Pac-Man's seek pellets behavior [118]

selector node shown in Figure 2.12 is a priority type which first attempts to find a ghost free corridor, if none are available, then it proceeds to check for a corridor with pellets. The last type of node in a BT is a decorator node, shown using the purple box. The decorator “adds complexity to and enhances the capacity of a single child behavior” [118]. Decorators can add conditional failures to child tasks, time limits for children to complete their task, or specify the number of times a child task should run. Terminal boxes, shown in white, contain action steps for the agent to carry out by the external domains if successful.

The taxonomy for software agents can be directly applied to system modeling, and behavior trees provide the logical breakdown necessary to model the internal domains. When modeling autonomy this taxonomy works fine, however, when attempting to understand the impact of automation, both the operator and technology are aggregated in a manner which hides the true system functionality. This aggregation is causing an unsustainable amount of uncertainty during requirement analysis and TIES as more automation technologies are evaluated and integrated.

Uncertainty

The approach of behavior trees in agent based modeling for system evaluations is already used in OR studies, however, it models the system as a whole without any attention to what subsystem is handling each behavior. Harper discusses some of the basic behaviors in his model: 1) “when the platform is within range of a ground threat, it reroutes to a safe area”, 2) “when the platform has a pre-determined amount of fuel remaining, it reroutes to return to base” [42]. These behaviors and this holistic approach may be fine when the impact of automation technology is not being evaluated. However, when trying to assess the technology, it is important to understand what the automation is handling and what the operator is handling: Who is deciding the route back to a safe area?, How is a safe area being determined?, If the technology is evaluating routes, does the pilot choose between a few routes?, What other demands are going on at the same time?. As shown above, the amount of workload and the awareness of the operator plays a direct role on the system’s effectiveness. The current modeling approach, which models the system as a whole, creates uncertainty during the evaluation of semi-autonomous systems because it is missing the dependencies between the operator and the technology. A holistic approach does not allow automation technology to be traded like other system technologies because the effects of the automation integration are not being captured on the operator or the system’s effectiveness. To discuss this uncertainty in more detail, first the types of uncertainty must be introduced.

When discussing uncertainty, there are two dimensions which it can fall in, aleatory and epistemic. Epistemic uncertainty is “any lack of knowledge or information in any phase or activity of the modeling process” [50]. This type of uncertainty may be mitigated through increasing the knowledge about the system. However, aleatory uncertainty is “the inherent variation associated with the physical system or the environment under consideration” [50]. This type of uncertainty can be better understood but not removed. The characterizing features of each type of uncertainty discussed by Fox & Ülkümen are shown in Table 2.3 [35]. Epistemic uncertainty is the result of inadequate levels of knowledge, therefore when

Table 2.3: Characterizing Features of Epistemic and Aleatory Uncertainty [35].

Feature	Epistemic	Aleatory
1. Representation	Single case	Class of possible outcomes
2. Focus of Prediction	Binary truth value	Event propensity
3. Probability Interpretation	Confidence	Relative frequency
4. Attribution of Uncertainty	Inadequate knowledge	Stochastic behavior
5. Information Search	Patterns, causes, facts	Relative frequencies
6. Linguistic Marker	“Sure”, “Confident”	“Chance”, “Probability”

it is removed the decision maker is 'confident' about that piece of information. However, aleatory uncertainty is about understanding the relative behavior of the system. The easiest example is presented by looking at a coin and a coin flip. The coin could be 50% likely to land on heads, or it could be weighted and 75% likely to come up heads; epistemic uncertainty would be not knowing that the coin is weighted, whereas determining whether a single flip will be heads or tails is aleatory uncertainty.

When looking at current modeling techniques, which capture the holistic system, epistemic uncertainty is the first to present itself. With no differentiation between information contained by/actions of the technology and that of the operator, modeling efforts provide inadequate knowledge to assess automation technologies. The decision maker has no ability to understand the interface in an operational scenario. In a larger context, aleatory uncertainty will also be present in properly capturing the operator's awareness, operator's workload, and the system effectiveness, because the operator adds a level of stochastic behavior. “The level of human behavior representation in [modeling] varies widely, but even the best of them assume ideal human behavior according to doctrine that will be carried out literally, and rarely take account of the vagaries of human performance capacities and limitations” [1]. To enable technology trades for automation technology, uncertainty surrounding the operator must be reduced. Although operator workload and awareness has not been explicitly modeled in OR system evaluations, it has been an increasing realm of interest.

Operator Workload and Awareness

The modeling of operator workload has seen a growing focus as issues pertaining to overwork and underwork persist regardless of technology integration. A 2018 paper by Hooey et. al states that “the modeling of workload in complex human–machine systems is a key element to supporting effective design and operations” [52]. Their work introduces a taxonomy for operator workload, breaking workload down into four primary drivers: environment, task, equipment, and operator. The classes and sub classes for this taxonomy are shown in Table 2.4. Starting with workloads dependence on the environment. When an operator’s visibility is decreased due to fog, obstacles, or darkness, they must work harder to acquire the same level of awareness. Similarly, environments with high levels of complexity such as dense traffic or a combat zone present higher levels of workload because they require greater levels of awareness to determine an appropriate action. Uncertainty and stressors (such as noise or g-force) have a similar effect on the workload. Task represents the second workload class. This class captures workload changes due to mission complexity/criticality (task demands), task duration/amount of information required (temporal demands), and multitasking (task structure). Variations in the task and timing that the task is received can have a significant impact on the overall workload of the operator. The third workload class is equipment. This class pertains to the other components of the system: vehicle, payload, and link. The amount of control required by the vehicle and the control authority of the operator, along with the number and type of payload that they are expected to control, can increase or decrease the workload experienced by the operator. When operator workload is modeled, changes in technology should be present in the number of actions required by the operator. Lastly, the operator himself provides some variation in the workload level. Increased proficiency may decrease the time to find information or change the action sequence, similarly, individual differences can change the level of workload due to difference in perception, cognition, or trust [52]. These classes provide a lenses through which technologies may be correlated to operator workload, while the operations

Table 2.4: Taxonomy for operator workload drivers, primary class and subclass [52].

Workload Driver Class	Workload Driver Subclass
Environment	Visibility Complexity Uncertainty Elements/Stressors
Task	Task Demands Temporal Demands Task Structure
Equipment	Vehicle Payload Link
Operator	Task Proficiency Individual Differences

model provides a good contextual environment to capture the effects.

2.2 Gaps

Based on the above discussion, analysis of systems which include automation technology requires a few additions to a traditional operations model. These additions are needed to reduce the uncertainty between the operator and the technology. There are six gaps which have been identified as necessary additions to facilitate operator workload analysis. The first three gaps represent a change of modeling technique when looking at a system as an intelligent software agent, while the second three focus on modifications to analyze a technology trade space. Starting with gaps in modeling, although the operator and vehicle are co-located, each component has both individualistic and overlapping components of a software agent (Figure 2.10). The vehicle has its own capabilities with regards to the external domains. A semi-autonomous system is capable of perceiving certain elements within its space, has communication capabilities, and has performance limits; however these are not equivalent to the external elements of the operator. The operator must be able to perceive

elements from his environment and the vehicle's displays before that information can be stored in the internal domains. This perception by the operator is workload on the operator as shown in the workload taxonomy, and must be monitored. If too many perception activities are required then the operator may suffer from over stimulation, however, too little and they may be unable to effectively manage the automation. Similarly, just because the vehicle may be capable of multi-channel communication the operator is limited by his senses. The operator is only aware of what he was able to ascertain. Lastly, the action set is different between the vehicle and the operator. The operator must seek knowledge and has management tasks, which are not captured by just modeling the vehicles actions. Similar problems exist when looking at the internal models of a software agent.

There are differences between the vehicle's and the operator's sense making, memory, and decision making. Sense making deals with data fusion; the vehicle has rigid procedures for utilizing multiple sensors and may be capable of taking in all of the information at once. Since the operator is comprehending a different set of information and has mental and physical resource limitations, he will be fusing a different set of data than the vehicle and may take longer to store this information in memory than the technology. The vehicle's memory capability now outpaces that of the operator. A multitude of information can be stored in the vehicle, however, this information takes an action to receive for the operator. The operator must decide his actions through his mental model. If time permits he may seek information from the system, however this acquisition will take time and create additional workload. The decision making in systems with low levels of automation is the sole responsibility of the operator, even though the mental models may be different. However, as the automation level is increased the system is capable of some decision making based on its memory, then the operator must decide when to intervene. These changes in internal and external features require changes to the agent formulation.

These differences mean that an agent that is meant to assess operator workload must have three additional features: sensory/SA model, dynamic task list, and a workload model

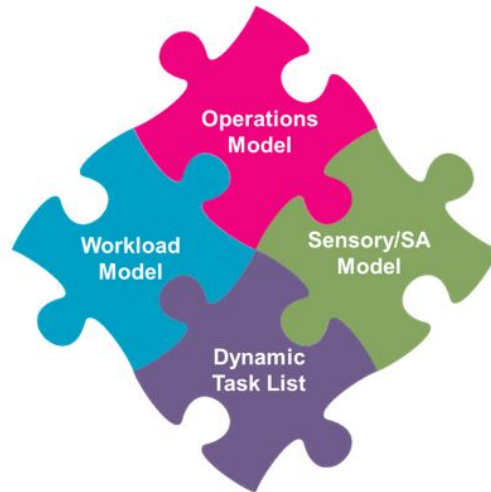


Figure 2.13: The three pieces that must be added to the operations model to reduce uncertainty around operator workload

(Figure 2.13). The sensory model is focused on ensuring that the proper memory construct is created for the software agent. Current technology studies use an omnipresent operator, if the vehicle is capable of sensing an environmental component then that information can be utilized in the decision making process. He is assumed to have the information and is able to utilize it whenever and however is required. This has been invalidated by the crashes of the Tesla and the gunner study. The vehicle is capable of acquiring much more information than the operator can utilize. Attaining each piece of information is associated with a time and workload, and the agent must go through sense making only with the information currently known by the operator. Once it is through sense making, this information is stored in the operator's memory (situation awareness) which can be utilized for operator-based decision making activities. The operator only receives updates to their memory stores when they decide to reacquire information through perception; he does not automatically refresh that information at some rate unless that is a component of his decision making.

Along the line of decision making, an operations model for assessing automation technologies requires an improved task list. Since current agent-based models focus on the system as a whole, they miss which tasks are done by the operator and which are done by the vehicle. The human-machine is modeled as one entity with the rule set defining the

combinations capabilities. Therefore, when technology is included, it is uncertain where the automation is impacting the system's operation. Behavior tree models must be detailed to separate automation from operator tasks. This provides clarity around the operator's function within the system and the number and type of operator tasks to be monitored throughout the scenario. The last addition is a workload model enabling the quantification of workload, a desire of OR studies. Currently, workload is set as an input, not measured as an output, thus it cannot be viewed or assessed early in the design process. Software agents that attempt to control the operator's workload do so through static limits on the number of tasks that the system is capable of simultaneously performing. Although this is a valid technique for automation, humans are not confined by static limits. It is better to understand the extent of tasks being requested of the operator, moving workload to an output where it can be assessed. The workload model looks to map the dynamic task list and sensory activities to a quantifiable and dynamic metric. Changing the sensory model, improving the task list, and quantifying workload in an operations model will enable OR studies capable of assessing workload.

With the OR model capable of assessing workload, it is important that it can be expanded to explore the technology trade space. This brings up the other three identified gaps, as introduced in the motivation section, pertaining to capturing system variations due to automation technology and providing useful information to decision makers. Gap 4 is focused on exploring the trade space. "Embracing a new era where 'joint warfare' means human-robot teams requires a better understanding of autonomy and a better effort to design for human-machine interdependence" [110]. This drives the need to be able to modify the three modeling components identified as gaps above in ways to capture numerous automation technologies. It is only through trade space exploration that this new methodology becomes useful in the development process. Once automation is included, the workload and awareness models must capture the level of trust between the operator and the technology. As noted by the NUWC study and shown in Figure 1.6, the level of

trust determines the frequency which an operator will double-check automated functionality. When the operator double checks information, they have a higher workload, but also higher awareness. Therefore, this effect and the resultant change in performance must be simulated within the OR model, representing gap 5. Lastly, gap 6 is presentation and utilization of this new information by a decision maker. This gap was identified in *Research Challenges in M&S for Engineering Complex Systems* and discussed above: “while there has been significant progress on understanding humans as decision makers, the utilization of this knowledge in M&S activities has been limited” [15]. OR studies are meant to inform, therefore this methodology uses M&S activities to inform the decision maker about the operator’s workload and resulting system performance.

2.3 Research Objective

The changes created by automation have created an unsustainable amount of uncertainty within the development process. The utilization of OR to understand the operational effectiveness of a design requires details around the operator workload. Designer’s must be able to understand the impact of automation integration, specify system requirements, and conduct performance trades based on this new information. Therefore, the formal objective for this research is to:

Develop a methodology that allows for the better assessment of system requirements and capabilities during early-design studies by decreasing the uncertainty pertaining to operator workload and enable technology trades based on cognitive factors, leading to enhanced system’s performance and operational awareness

It is important that the information provided is useful in the decision making process early in the development cycle. The new methodology must enable the assessment of a diverse set of technologies to facilitate trades and the output should enable the creation of

good system requirements. The desire to assess different automation technologies requires a re-configurable software agent, which can quickly capture the impacts on the operator's workload. Due to cost and time constraints, these technologies will be evaluated using OR, which requires a dynamic methodology that is capable of being captured in a simulation environment. Lastly, the methodology must inform system requirement creation. Discussed above, the DoD defines a good requirement as: achievable, verifiable, unambiguous, complete, written in performance terms, consistent, and providing the appropriate detail. Many of these criteria do not factor directly into the methodology, however, verifiable, performance terms, and appropriate detail are noteworthy because they reinforce the need for the methodology. The quantification of workload allows new requirements to be created based on desired operator workload levels. The addition of new automation can be verified against measured workload levels to ensure that they are acceptable. Traceability between the tasks and the measurement capture the operator's actions leading to the specified system performance. Lastly, the appropriate level of detail constrains the solution to only use information available during early phases of design. This methodology focuses on operator workload without getting down to the details of system layout.

2.4 Research Problem

Based on the research objective, the goal is to create and evaluate a methodology that decreases the uncertainty pertaining to operator workload in partially-automated system analysis. The research is broken down into four primary stages, shown in Figure 2.14. The first stage captures the operator in an agent-based environment and defines the operator workload metrics. The findings from this stage will be compared to literature to determine the validity of the methodology. The second stage assesses the necessary changes to the agent-based model to capture different automation technologies. The third stage focuses on the utility and visualization of the new data to inform decision makers. These three stages are important to defining the methodology. The agent creation and result verification

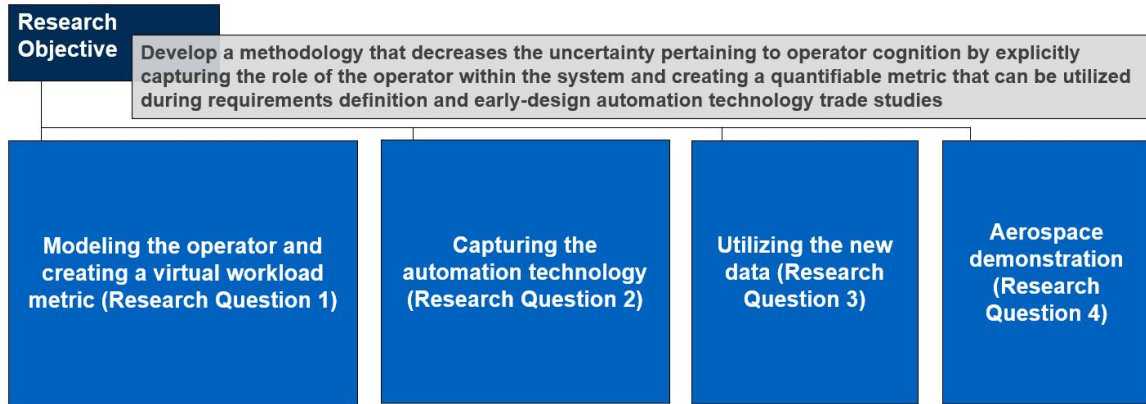


Figure 2.14: Overview of the four stages to create and evaluate a methodology that decreases the uncertainty pertaining to operator workload in partially-automated system analysis

require data from literature to ensure alignment between the virtual model and real-world experimental data. Therefore, these stages are done in the automotive space. After applying the methodology to an environment with verification data, the final stage of the research demonstrates how the methodology can be used in an aerospace application.

The methodology defines a process for capturing the role of the operator in a virtual environment. This work is intended as an approach for better defining models within the operations research (OR) process; it is not intended as a replacement of the current process. Therefore, the three phases of OR (formulation, analysis, interpretation) are utilized to breakdown the automation environment and define the elements of the methodology. As a reminder, the OR process is shown in Figure 2.7).

2.4.1 Formulation

The first step in formulation is orientation. It is important to understand where and how the technology is impacting the operator and how it is being studied. Studying the impact of technology on the operator is often a manual process using human-in-the-loop methods. This type of study is shown in the work of Edwards et. al which looked at situational awareness and workload for an air traffic controller (ATC) with different levels of technology [27]. The study looked at the introduction of technologies and the impact this had on

operations [27]. The ATCs were tested under four conditions of automation with workload levels self-reported. The summary of these conditions and the workload levels are shown in Table 2.5. The first condition automated all tasks, the operator was solely responsible for identifying conflicts. Under the second condition, the operator identified conflicts and handled his routine tasks such as pilot check-ins and aircraft hand-offs in and out of the sector. The third condition replaced routine tasking with decision making, which forced ATCs to handle requests from flight crews in coordination with adjacent sensors. While the fourth condition had ATCs handling all three components. The perceived workload under each condition is shown in Table 2.5. The condition with the highest level of automation (condition 1) showed the lowest rated workload, with workload increasing as automation decreased. However, of interest is also the system's performance to detect conflicts. Under all conditions, ATCs were required to enter a keyboard command when they detected a conflict. The time taken to detect conflicts under the different conditions is shown in Figure 2.15. This figure shows that although the workload was lowest for condition 1, it also took the longest for the operator to recognize conflicts. The quickest to detect conflicts was condition 2, which automated the decision making component but left the ATC to handle his routine tasks and detect conflicts: "on average, participants responded to SA questions more slowly in the most automated condition (condition 1, CD only) and condition 3, CD + DM, suggesting reduced SA" [27]. Studies like this show the dependence between technology integration and the workload/situational awareness of the operator. However, since these studies require human-in-the-loop testing, they are time and labor intensive, and are typically carried out later in the development process when fewer design freedoms remain.

Some studies have attempted to model pilot workload. An example of this type of work is shown by Schneider et. al who simulated pilot workload for the control of multiple unmanned aircraft systems (UAVs) [93]. Their study focused on the need for an executable architecture and enabling workload as a measure of performance for the system. They used the experiences and discussions with MQ-1Bs pilots to set a workload limit, then analyzed

Table 2.5: Self-reported average workload ratings by condition [27].

Condition	Mean	SD
1) Conflict Detection (CD)	3.09	0.54
2) CD + Routine Tasks (RT)	3.46	0.74
3) CD + Decision Making (DM)	3.39	0.46
4) CD + RT + DM	3.71	0.76

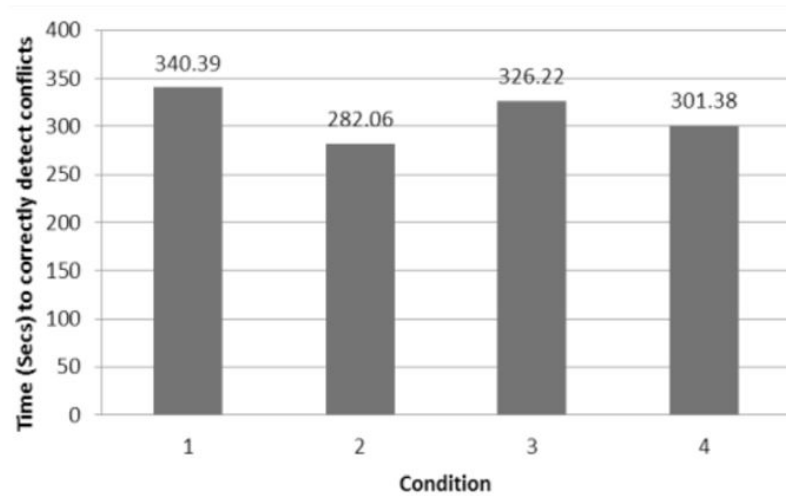


Figure 2.15: Results from Edwards et. al showing the time taken by an air traffic controller to detect conflicts under different levels of automation (conditions detailed in Table 2.4) [27]

the inclusion of UAVs to determine the frequency which this limit was exceeded. Figure 2.16 shows the measured workload for a pilot controlling a single UAV performing an ISR mission with the presence of a dynamic event compared with the same scenario but the pilot is now performing multi-aircraft control (MAC) utilizing four UAVs. For the MAC case, the operator was above the workload threshold for 29.3% of the mission with a workload peak of 351, compared with 2.6% and just above 100 for the no MAC case [93]. It is noteworthy that both cases went above the workload threshold, however since the no MAC case was above only for a brief duration it represents a difficult but manageable workload according to the MQ-1 pilots [27]. This emphasizes the importance of understanding the workload in an OR setting where. Workload can be assessed in context with soft limits on an operator's cognitive capabilities. This study is closer to enabling automation technology assessments, however, in its current form it is still lumping together operator workload types, thus the primary drivers of workload are ambiguous. This study also ignored the implications of automation on the operators situational awareness thus possibly mission success.

Orientation to the problem shows that there are studies which assess the operator's awareness and workload, however, this is an often manual process thus not dynamic for early design studies. There are a few cases which analyze operator workload in simulation, however, they ignore the coupling between workload and SA on mission effectiveness and do not currently provide the tracibility required for requirements definition and trade space studies. Therefore, moving to the OR problem formulation stage, this research focuses on the quantification of operator performance in a dynamic environment to define requirements and answer technology integration questions. The assessment must provide three capabilities, which were originally introduced in Gaps and are captured in Figure 2.13. First, it must allow clear task assignment between the operator and the automation. This task assignment is a primary driver of workload and situational awareness as shown by Edwards et. al [27]. Second, the operator's awareness must be captured throughout the

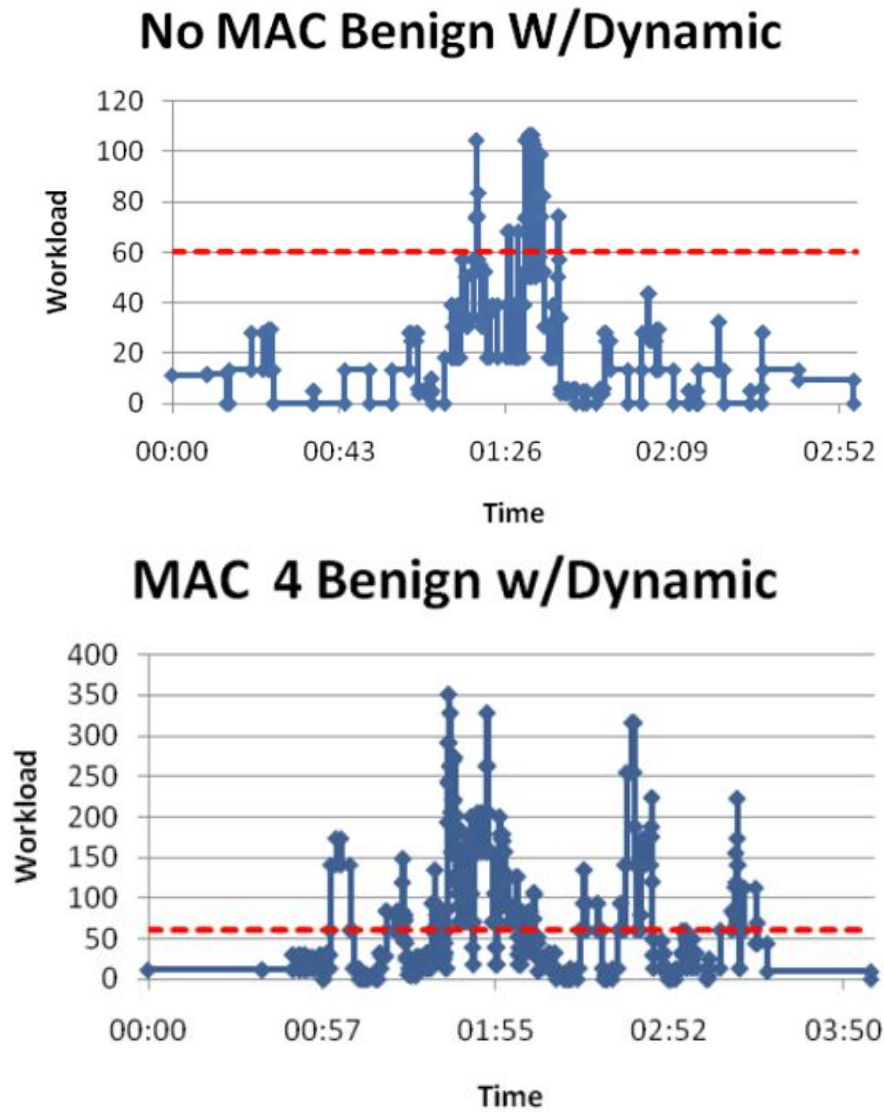


Figure 2.16: Results from Schneider et. al showing modeled workload levels for and ISR mission with and without multi-aircraft Control (MAC) [93]

scenario to properly capture system effectiveness. Edwards et. al [27] showed that lower levels of awareness are tied to poor performance. Lastly, the cognitive workload must be provided as an output such that it can be assessed for areas above or below acceptable baselines. Schneider et. al [93] showed that the inclusion of additional task loading from technology may drive unsustainable amounts of workload.

The last step in formulation is data collection. Although the other steps may be generalized for the creation of this methodology, data collection is more difficult to discuss in a general context. In many ways, data collection for this problem is combined with model formulation because the model will be created based on task flows, evaluation times, and workload mappings. These pieces of information must be collected, however, there discussion will be delayed until the model formulation is introduced.

2.4.2 Analysis

The next step in the OR process is model formulation, which starts the analysis phase. However, before discussing the formulation it is important to introduce a new engineering field. Systems design and operations research were introduced and discussed in the background. The overlap of these two disciplines has already occurred and continues to grow. Together these disciplines are capable of providing performance trades for systems in an operational scenario and create vehicle centered performance requirements. However, as the interest shifts to understanding the operator, it is important that the field of cognitive engineering is introduced. Cognitive engineering or cognitive systems engineering (CSE) is “the study of cognitive work in context for the purpose of improving system effectiveness and the safety and productivity of the human constituents of the system” [20]. It is a relatively young field, growing in prevalence after the 1979 accident at Three-Mile Island. This incident drove a need to understand patterns and principles in human-computer interaction and support the design of human-computer interfaces. CE growth has been driven by three forces: 1) growing complexity of socio-technical systems, 2) problems and fail-

ures created by a clumsy use of emerging technologies, and 3) limitations of linear models and the information processing paradigm [51]. For more detail about the driving forces see Hollnagel and Wood's book, both recognized experts in Cognitive Systems Engineering, *Joint Cognitive Systems* [51], which spends the first chapter on driving forces. CSE is able to enhance systems design through the inclusion of its dedicated study of operator cognition in context.

CSE has been broken into many disciplines. Based on the OR problem definition, two have been identified as focus areas: situational awareness and hierarchical task analysis. Situational awareness (SA) is “an understanding of the state of the environment (including relevant parameters of the system). It provides the primary basis for subsequent decision making and performance in the operation of complex, dynamic systems” [28]. Research in this area has focused on understanding and assessing how operators gain, evaluate, and make decisions based on their SA. This will provide the necessary information and frameworks to change the software's perception, sense making, and memory elements to behave in a manner that is more representative of a human operator. Hierarchical task analysis (HTA) is one of the most popular human factors and cognitive engineering models [91]. It is focused on capturing the actions taken by an operator to achieve his goal (an objective or end state), therefore task in this usage is the problem that the operator is trying to solve not the individual actions determined during the breakdown [91, 57]. HTA provides the information to better define the decision making and action elements within a software agent.

This introduction to CSE brings up the first driving research question:

Research Question 1.0: Can the inclusion of cognitive engineering disciplines in the operational assessment of systems reduce the uncertainty pertaining to the operator's cognitive load by providing a dynamic, traceable, and quantifiable measurement throughout the scenario?

The notional overlap of these fields is represented in Figure 2.17. Cognitive analysis

and system design are already integrated in many instances, with cognitive engineering driving design decisions, especially later in the design process. However, to assess technology CSE must also be better integrated with OR modeling to capture cognitive effects early in the design process. CSE helps decrease the uncertainty pertaining to the operator's cognitive factors by providing disciplined research that facilitates the creation of improved software agents. These agents capture the vehicle-operator interaction and change the system performance accordingly. This better agent creation allows cognitive factors to be assessed during the requirements definition phase by moving workload from an input to an output, and allows decision makers to understand the implications of automation integration on mission measures of effectiveness. With this improved understanding, automation technology trades can be conducted in the same way that traditional system characteristics (engines type, materials, airfoils, etc) are traded. The necessary components for model formulation will now be discussed in more detail, utilizing the components of HTA and SA.

Dynamic Task List

Through gap analysis, three components were deemed necessary additions to an operations model to facilitate automation technology trades. The first of these is a dynamic task list, which attempts to solve the gap relating to modeling at the whole system level versus capturing the vehicle-operator interaction. Currently, agent-based models for systems analysis focus on the capabilities of the system, missing out on the goal-driven and capability limited actions that the operator may take based on his awareness level. This brings up the first sub question relating to the assessment of cognitive load:

Research Question 1.1: *Can a dynamic task list be created from the operator's perspective to map operator awareness to a goal-action agent-based rule set?*

To answer this question, more information must be provided around hierarchical task analysis (HTA). As briefly introduced above, HTA provides a methodological breakdown

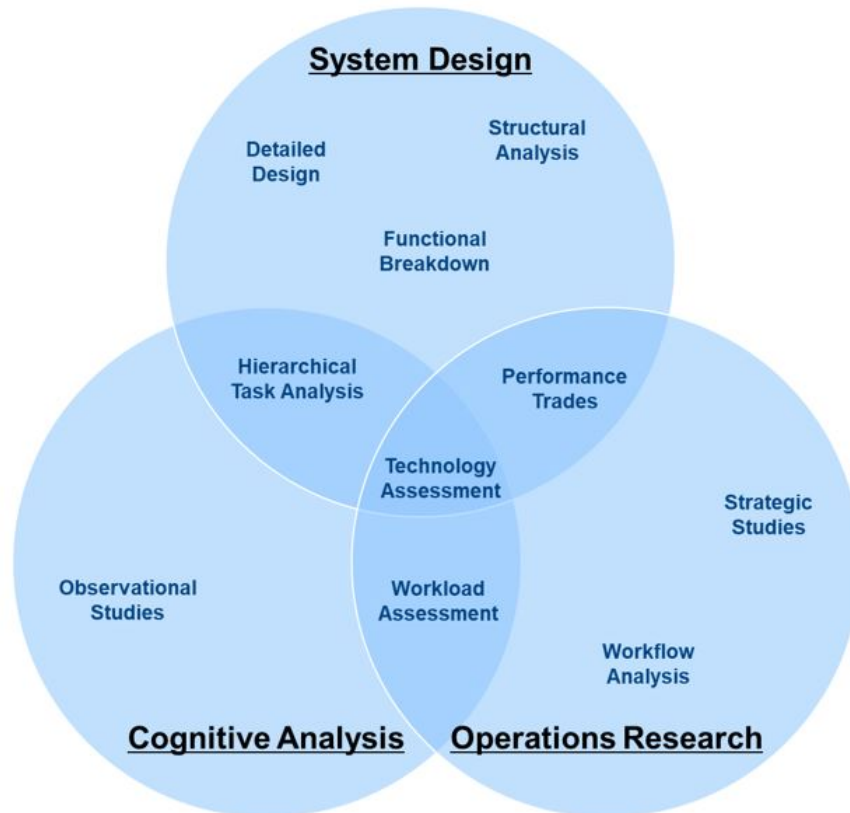


Figure 2.17: The notional relationship between the fields of system design, operations research, and cognitive analysis to provide technology assessments

of an operator's actions and sub states to achieve a desired end state. It was developed in the 1960s to better understand complex cognitive tasks and can be applied to either vehicles/technology or operators. Stanton [102] discusses the procedure for breaking down the sub-goal hierarchy, with his flow chart shown in Figure 2.18. Since it is typically easier to discuss the process with an example, a hierarchy from Li's work [64] for the go-around procedure of a large commercial aircraft is shown in Figure 2.19. The HTA process starts with the identification of the overall goal, in the case shown it is performing a go-around procedure but in a general aircraft case it may be a safe flight with go-around being a sub-goal of landing (which would be a sub-goal of safe-flight). Once the overall-goal is stated the first subordinate goal should be defined. Goals have three components, although sometimes some of the components are not required. A goal has an activity verb which

defines the objective, performance standards by which the goal will be measured, and any conditions under which the goal must be performed [102]. The next step in creating an HTA is defining the state plan. This step specifies how the sub-goals will be handled by the subject of interest. Types of plans include: linear, non-linear, simultaneous, branching, cyclical, and selection. The sub-states of go-around follow a linear plan, with each sub-goal needing to be achieved before proceeding. HTA is an iterative process so as detail is added, the analyst should ensure that the description is appropriate based on the desired results of the study. This is the step where a sub-goal could be broken into further sub-goals as deemed appropriate. Once a state is defined this process continues until the analyst is comfortable with the breakdown, at which point the breakdown created with HTA should be discussed with a subject matter expert [102].

The relationship between this methodology and the implementation of behaviors trees in software agents is apparent. The HTA provides a methodology for more accurately representing state transitions based on operator studies and agreement with subject matter experts. However, in the current form the HTA provides desired states but cannot drive the agents action, and does not discuss operator awareness considerations. Therefore, the additional step of adding operational event sequence diagrams (OESDs) using the HTA is required. “To construct an OESD, an analysis of the task sequence [is] required together with and understanding of how each operation was performed and by whom (or what)” [46]. An OESD for changing lanes in an automobile created by the National Highway Traffic Safety Administration (NHTSA) is presented in Figure 2.20. The OESD shows how the subject vehicle (SV) goes through the state ‘changing lanes’, while tying in the internal domains memory and the external domain of perception. The OESD displays the information-action-decision sequence which can be utilized in the definition of a software agent. This particular example shows the depth of sequence that the OESD can describe, however, not all OESDs are at this level and they can be tailored to the application. This level of detail around the operator tasks as a function of smaller actions which are conducted

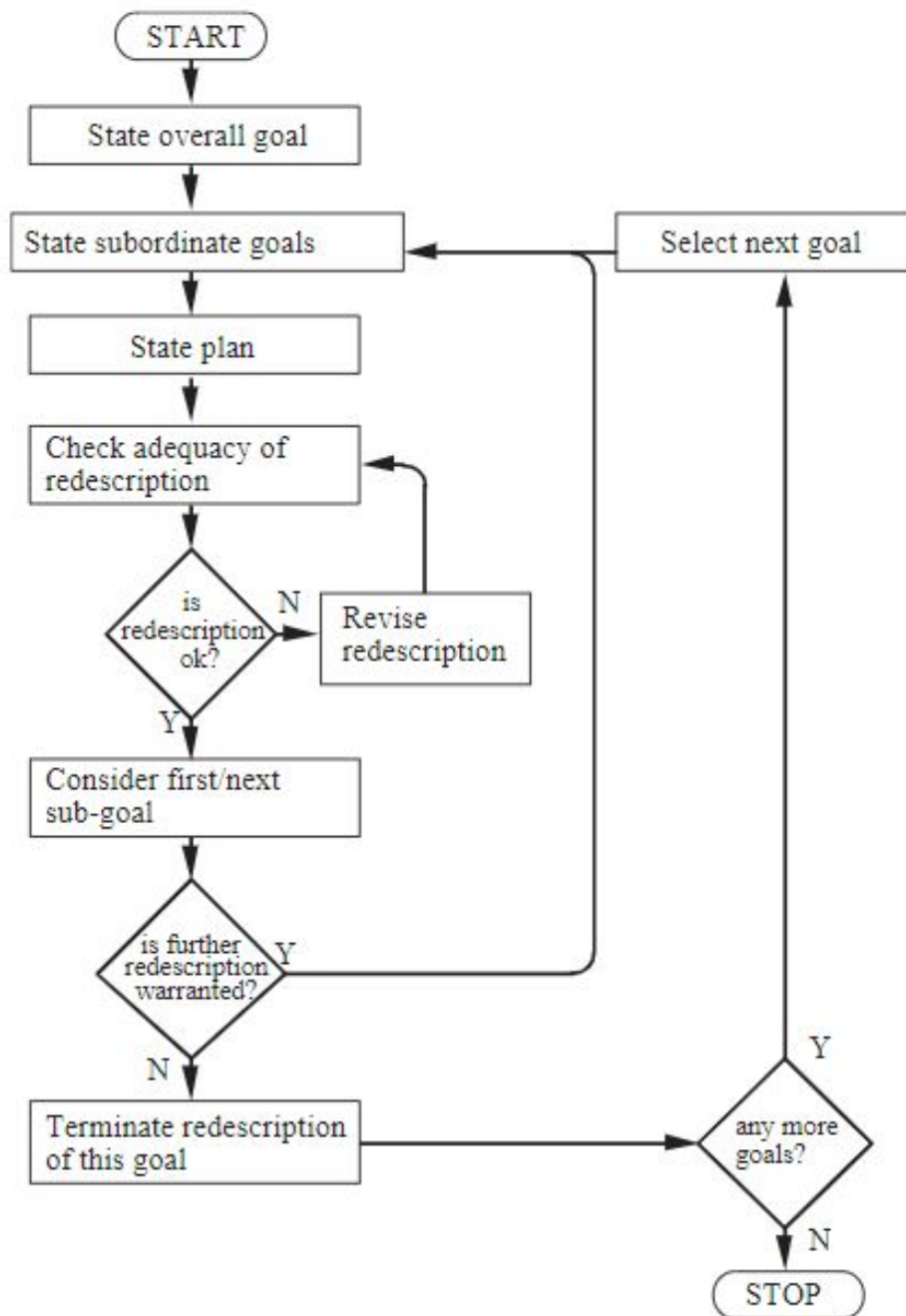


Figure 2.18: Flow diagram representing the steps to conduct a hierarchical task analysis [102]

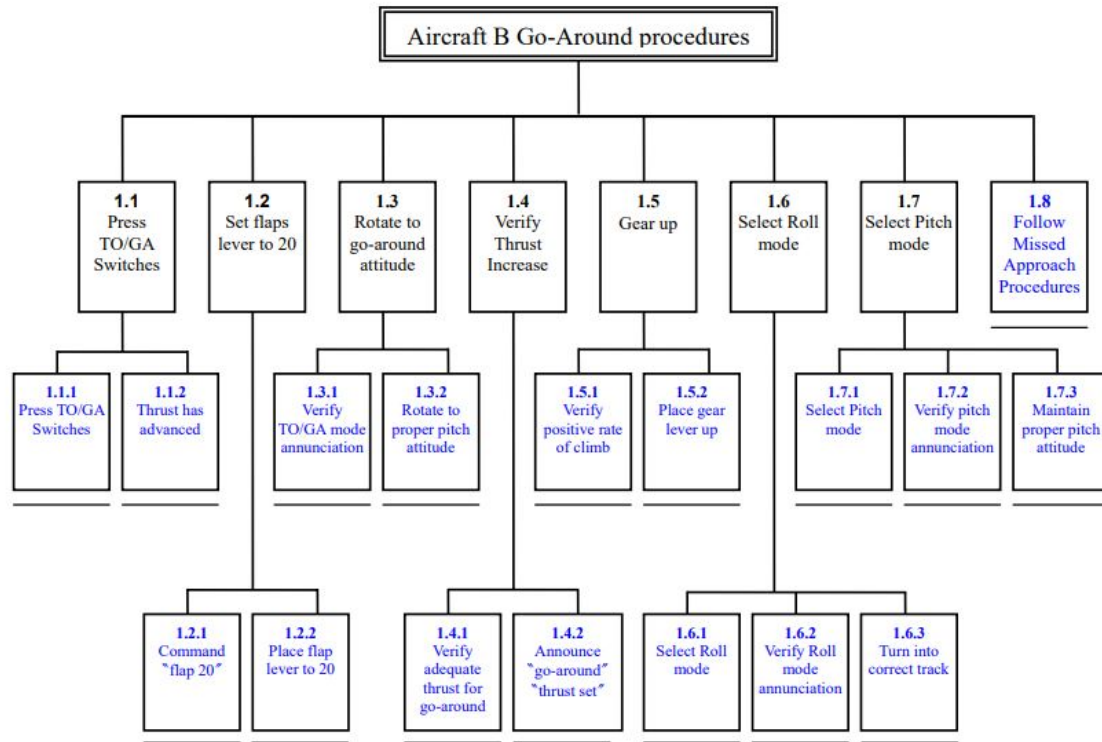


Figure 2.19: Procedure breakdown for an aircraft performing a go-around using a hierarchical task analysis [64]

based on awareness brings up the first assertion:

Assertion 1.1: utilizing HTA and representing goal-action sequences will enable the creation of a dynamic task list that can create an agent rule-set capable of guiding behaviors based on operator awareness

HTA provides the goal oriented approach which allow changes to the task list as new automation concepts. The high level goal will remain the same, however, the sub-goals will change. This change in goal structure will similarly change the OESD. These changes can be implemented as modifications to the software agent's rule-set to capture the effect of automation changes in an OR study.

Sensory/Situational Awareness Model

The OESD shows the connection between awareness and action sequence. Therefore, it is important to improve the representation of the operator's memory, not just utilize the

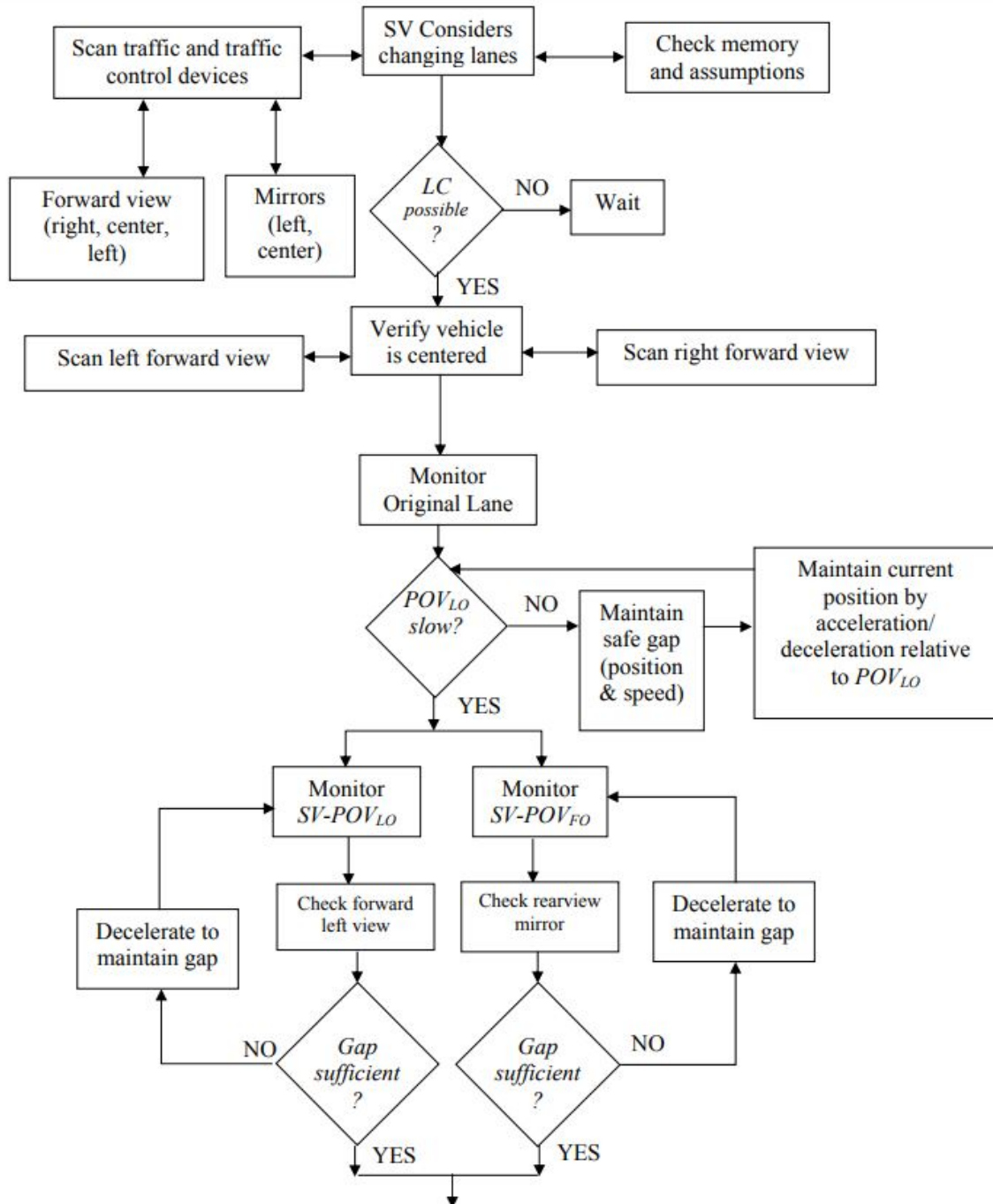


Figure 2.20: NHTSA operational event sequence diagram (OESD) for changing lanes [62]

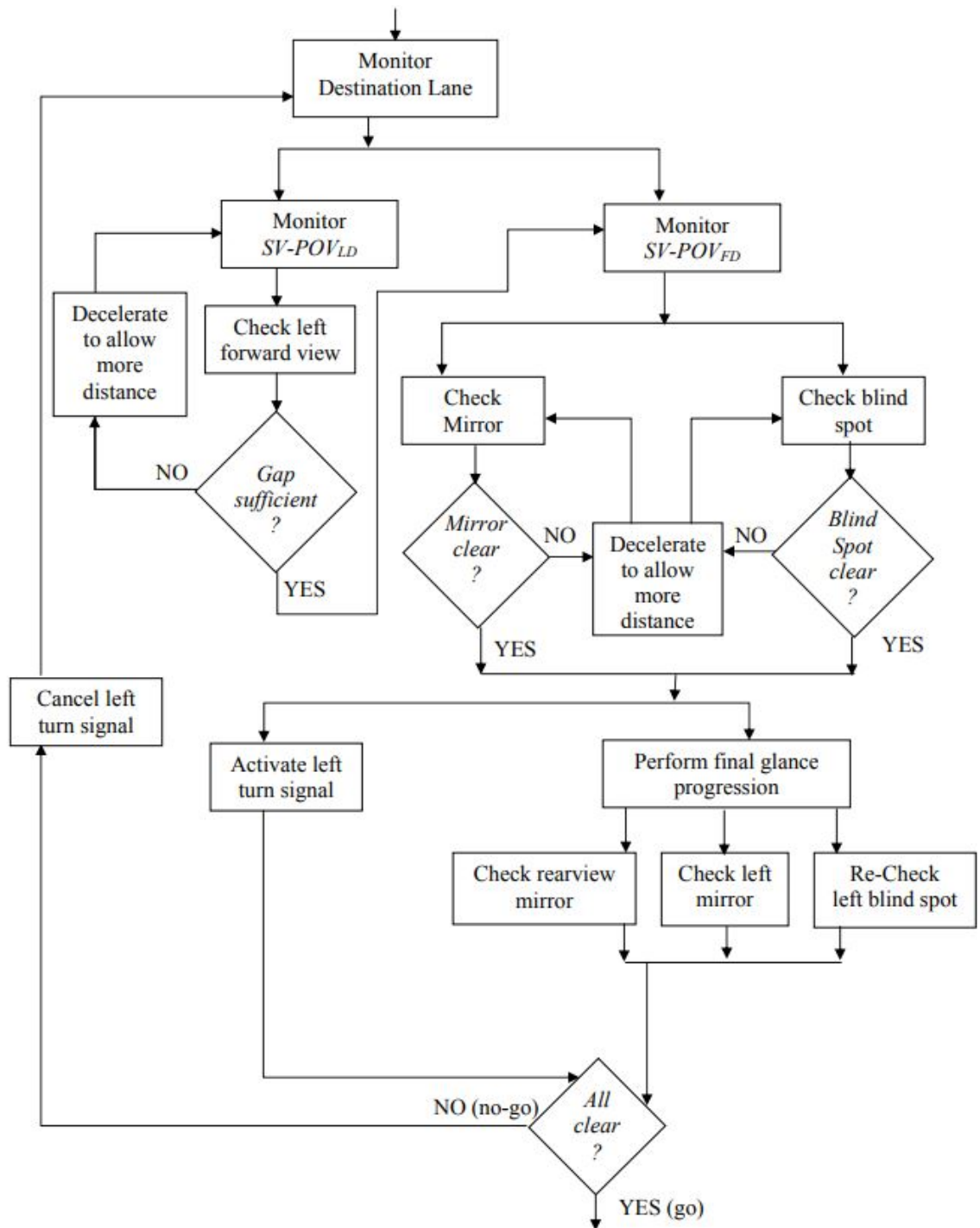


Figure 2.20: NHTSA operational event sequence diagram (OESD) for changing lanes (cont.) [62]

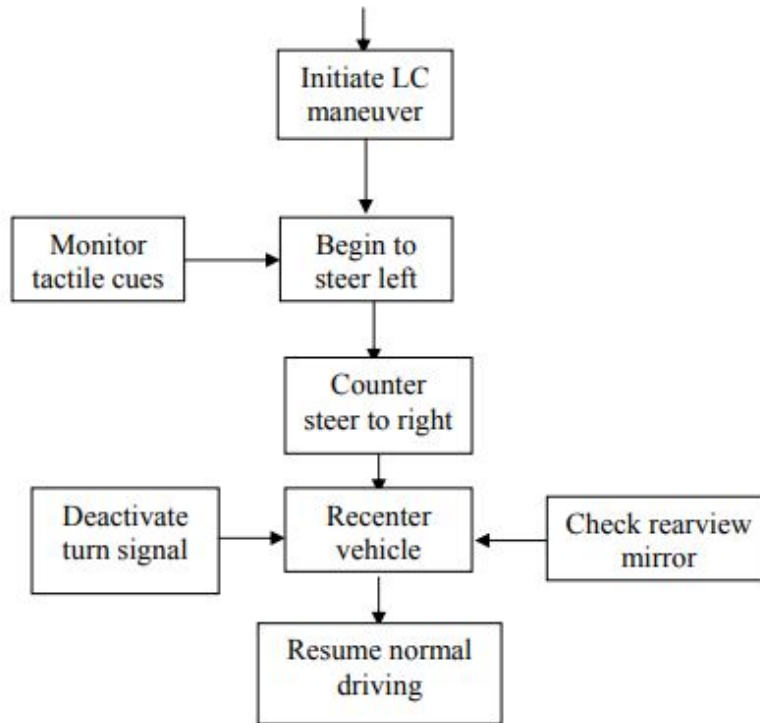


Figure 2.20: NHTSA operational event sequence diagram (OESD) for changing lanes (cont.) [62]

system's capabilities. The second gap focused on users being omnipresent, if the system is capable of perceiving an object then the operator is capable of using that information, however, the studies discussed above show the effects of automation integration on an operator's SA and the uncertainty introduced by modeling the operator as omnipresent. Therefore, the next research question pertains to acquiring and utilizing environment information:

Research Question 1.2: Can the operator's knowledge acquisition and information utilization processes be explicitly modeled to include acquisition delays and utilization challenges which directly impacts awareness and how the agent follows its dynamic rule set?

To answer this question, it is important to further discuss the field of situational awareness and the models and techniques used to understand operator SA. The field has grown since the 1980s but it came to prominence in 1995 with a special edition of the Human

Factors Journal focused on Situational Awareness. Endsley, one of the most referenced authors in the field, defines SA as “the perceptions of the elements in the environment within a volume of time and space, the comprehension of their meaning and the projection of their status in the near future” [29]. Figure 2.21 shows Endsley’s 1995 SA model, which continues to be the most referenced, depicting how these levels of operator situation awareness build upon each other and how SA relates to decision making. The first step in achieving SA is the operator’s perception of “the status, attributes, and dynamics of relevant elements in the environment” [31]. The determination of relevant elements is based on the current mental model and goals of the operator. Once an element is perceived, it must be comprehended. Comprehension is the effort required by the operator to convert and combine perceived elements into meaning that he can utilize to make better decisions. In example, the time taken to understand that a squadron of aircraft may be a bigger threat than the pair, or combining warning lights to diagnose a system problem. The final level and highest level of SA is projection. This level of awareness describes an operator who can utilize his current understanding of the environment to project future actions. Going back to the squadron example, a pilot may use heading, location, and type information to attempt a specific intercept to maintain a tactical advantage. Overall, “the three levels of SA represent ascending levels of SA, not linear stages...Based on [the operator’s] goals and current understanding and projections (Level 2 and 3 SA), [he] may look for data to either confirm or deny [his] assessments or to fill in gaps” [32]. The 1995 Endsley model presents a framework through which to view SA and relates its importance to the decision making process in a feedback environment.

This three level definition is important because operators take time to acquire and utilize information from their environment. The level of automation changes the type of information that the operator is receiving and the decision that he is expected to make. Based on this change in effort and understanding, the operator will have different workload levels and may make a different decision because information may be more or less apparent. The

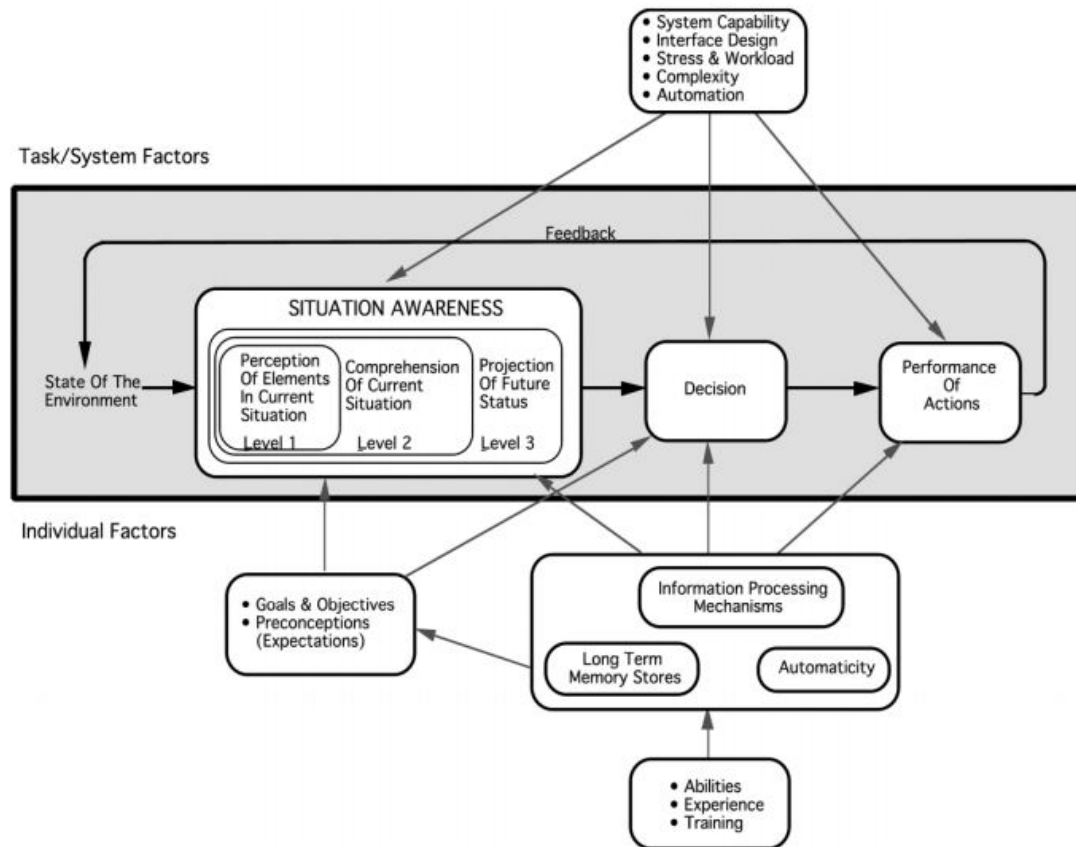


Figure 2.21: Endsley's model for situation awareness in dynamic decision making [31]

performance of the system is dependent on this decision making process. Therefore, there are two factors which must be improved to accurately portray operator SA in a software agent for technology evaluations. The perception domain must be improved to capture the time taken by an operator to understand elements in his space and the mental model must change as higher levels of SA are acquired to guide better decision making. The perception model must account for the difficulty of knowledge acquisition and time required to acquire information. A primary driver for the level of situation awareness and related workload is the complexity of the environment, introduced above in the taxonomy of workload and discussed by Hooey et. al [52]. Figure 2.22 shows three different driving situations which would affect the workload and possibly impair driving through variations in decision making [21]. This figure introduces three different situations (simple, moderate, and extreme) which would describe environments that one driver may interact with. Going

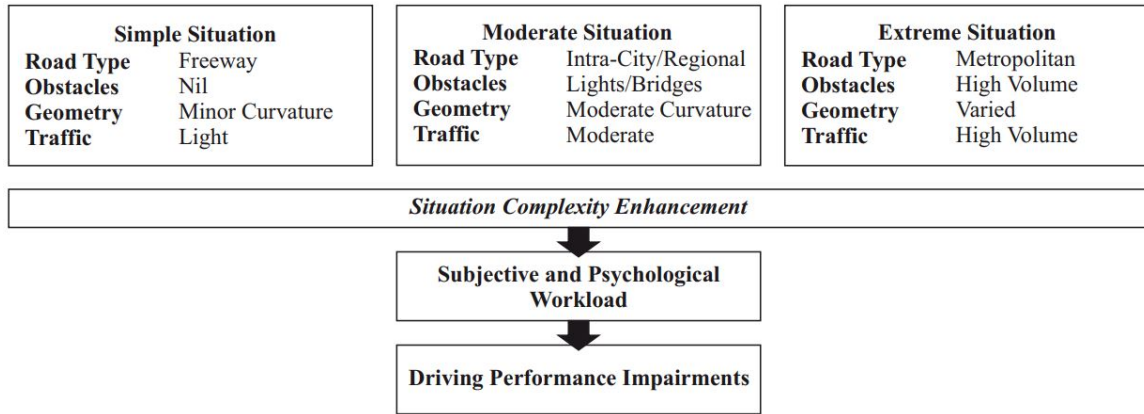


Figure 2.22: Dependence between environment complexity, operator workload, and system performance [21]

up to the operational event sequence diagram (OESD) discussion and Figure 2.20, it can be seen how these environmental factors play a role. Tasks such as checking a blind spot or forward-left view are of greater difficulty in environments which have varied geometry roads and high traffic than in environments with minor curvature and light traffic. Agent-based modeling can provide the variations in situation, however, the perception domain on the software agents must vary their capabilities accordingly. The variations can be driven by SA studies and subject matter experts on information acquisition time and types. Studies have been conducted to understand an operator's acquisition process during a lane change event, "mirror glance times range from 0.8 to 1.6 s...for a sedan lane change from right to left, the probability of glancing at the forward view was 0.41, the probability of glancing at the left mirror was 0.22, the probability of glancing in the center mirror was 0.21, and the probability of glancing over the left shoulder (blind spot) was 0.08" [62].

These times and probabilities are captured through human based studies, either simulated or observational. There are many techniques currently utilized within the field of SA. It is not proposed that this current methodology would replace any of these measurement techniques. This process must leverage the data gleaned by these studies and fill in missing information with subject matter expertise to more accurately represent the software agents. The studies in the field of SA provide information about times to acquire information, the

probability of acquiring that information, and a framework for modifying the mental model of the agent, leading to the second assertion:

Assertion 1.2: the studies conducted in the field of SA will allow for the dynamic representation of the costs of information acquisition and the challenges with information utilization which can be utilized in the activation sequence of the dynamic task list

The probabilities and times modify the perception domain, while the SA framework specifies modifications to the internal domains. Higher levels of SA require the sense making internal domain to correlate pieces of information then store this in memory. These higher levels of SA take greater bandwidth and time: understanding the expected next actions of a neighboring agent takes longer than just determining its presence. Unlike a physical sensor on the system that receives information in a very deterministic manner, an operator takes time to maximize his understanding. If the task list forces a quick decision then the operator will be forced to act on a lower level of SA than if he has more time.

Workload Model

The last component of interest is the dynamic measurement of the workload induced by acquiring SA and the task load. There have been many workload studies as seen by the works discussed above [13, 12, 98, 52, 27]. However, many of these are manual, human-based studies which are resource intensive and difficult to utilize early in the development process, and none of them are capable of technology trade space exploration. This brings up the last sub-question for Research Question 1:

Research Question 1.3: Can the typically manual, human-based workload measurement techniques be applied in an agent-based environment to provide a quantitative assessment of workload?

Reducing the uncertainty around mental workload requires a model that can move operator workload from input to dynamic output based on the current environment and vehicle demands of the software agent. Mental workload is “a product of the demand/s of the task and the capacity/ies of the person performing the task, where demands and capacities may be moderated by context” [75]. This research will focus on the mapping of demands, from perception and task activities, to workload. There are three primary categories which capture the many mental workload measurement methods: 1) self-report, 2) behavioral secondary tasks, or 3) physiological measurement [75]. Self-reporting is most often used, as it is the cheapest and easiest to implement. The most commonly referenced tool is NASA’s Task Load Index (TLX), which is a subjective measurement method that an operator uses to assess his workload level throughout a specified task.

The NASA TLX uses a questionnaire to assess an operator’s workload among six rating scales, as defined in Table 2.6 [47]. The scales of interest for this study are the mental and temporal demands, although the methodology could probably be expanded to also capture the physical demands. Mental demands will capture the number of tasks required by the operator, while temporal demands will focus on how often an operator’s decision was forced before desired levels of SA could be acquired. Performance is relevant but will be captured using traditional measures of performance rather than trying to capture the operator’s perspective (just because the software agent may be unaware that the mission failed, the performance will note the failure). Effort and frustration will be ignored for the time being since they are more subjective and rely heavily on the interface design along with the automation functionality. NASA’s TLX scale gives a framework for capturing cognitive workload and its prevalence in research provides studies for comparison, however, its subjective nature does not provide the numerical basis needed to output the workload.

To understand task-based workload, it is important to look at physiological measurement techniques which provide an objective basis. Studies have focused on monitoring a participant’s physiological behavior during task performance and correlated those results

Table 2.6: Definitions for the six rating scales used by NASA’s Task Load Index (TLX) [47].

Scale Title	Endpoints	Descriptions
Mental Demand	Low/High	How much mental and perceptual activity was required (e.g., thinking, deciding, calculating, remembering, looking, searching, etc.)? Was the task easy or demanding, simple or complex, exacting or forgiving?
Physical Demand	Low/High	How much physical activity was required (e.g., pushing, pulling, turning, controlling, activating, etc.)? Was the task easy or demanding, slow or brisk, slack or strenuous, restful or laborious?
Temporal Demand	Low/High	How much time pressure did you feel due to the rate or pace at which the tasks or task elements occurred? Was the pace slow and leisurely or rapid and frantic?
Performance	Good/Poor	How successful do you think you were in accomplishing the goals of the task set by the experimenter (or yourself)? How satisfied were you with your performance in accomplishing these goals?
Effort	Low/High	How hard did you have to work (mentally and physically) to accomplish your level of performance?
Frustration Level	Low/High	How insecure, discouraged, irritated, stressed and annoyed versus secure, gratified, content, relaxed and complacent did you feel during the task?

with performance. Some of the techniques utilized are monitoring brain wavelength, monitoring the brain oxygen and glucose levels, and eye tracking studies [75, 69]. An example of this type of study is shown by the work of Martin et. al, which used eye-tracking to assess the mental workload of an air traffic controller performing a task with and without conflicts [69]. Their study described how the conflicts experienced by the controller increased his mental workload. Eye tracking measures eye movement and pupil dilation during task performance to assess workload and fatigue levels. This methodology requires an operator (volunteer) in the loop to conduct the analysis and the resources required do not lend it to technology trade studies. It has been primarily used within design to provide

dashboard analysis and information presentation assessments, see *Engineering Psychology and Cognitive Ergonomics* [45]. However, studies like this can provide an objective basis to help correlate the dynamic task list to workloads scales within the NASA TLX.

The implementation of workload analysis in a virtual environment can be seen in the U.S Army's Improved Performance Research Integration Tool (IMPRINT) Pro, which has been accepted as the tri-service tool for workload analysis. IMPRINT uses task-level information to construct and parameterize networks representing the flow, performance time and accuracy for operational or maintenance missions [44]. The primary elements of an IMPRINT model are shown in Figure 2.23. There is a variable set which represents the system, these variables are changed based on the scenario events, and, based on the state of the variables, a dependent task network is executed. IMPRINT uses Micro Saint Sharp as its engine, an embedded discrete event task network modeling language. A sample task network flow is shown in Figure 2.24. This sample flow describes the tasks involved in a tank tactical march back to an assembly area, so each discrete event represents a movement if possible. Inside each task block in the network diagram exists a set of operator tasks that must be conducted. An example is shown in the analysis tree, i.e. adjust the interior requires the operator to adjust his personnel heater. This operator task is broken down into taxons, shown for a maintenance task in Figure 2.23. Each taxon represents an associated workload on a specific factor. In example, inspection creates visual and cognitive workload, while lubricate is a function of fine motor and visual workloads. IMPRINT's reliance on a discrete event simulation requires an explicit definition of the scenario, so it does not enable the dynamic mission assessment desired for trade space exploration. However, the framework does provide a basis for simulation based operator workload analysis.

This methodology is reliant on an effective workload measurement method which can be implemented in an agent-based simulation. The method must satisfy the dynamic and traceable requirements discussed above. The hypothesis for Research Question 1.3 is:

Hypothesis 1.3: *The detailed task list will map to different elements of the*

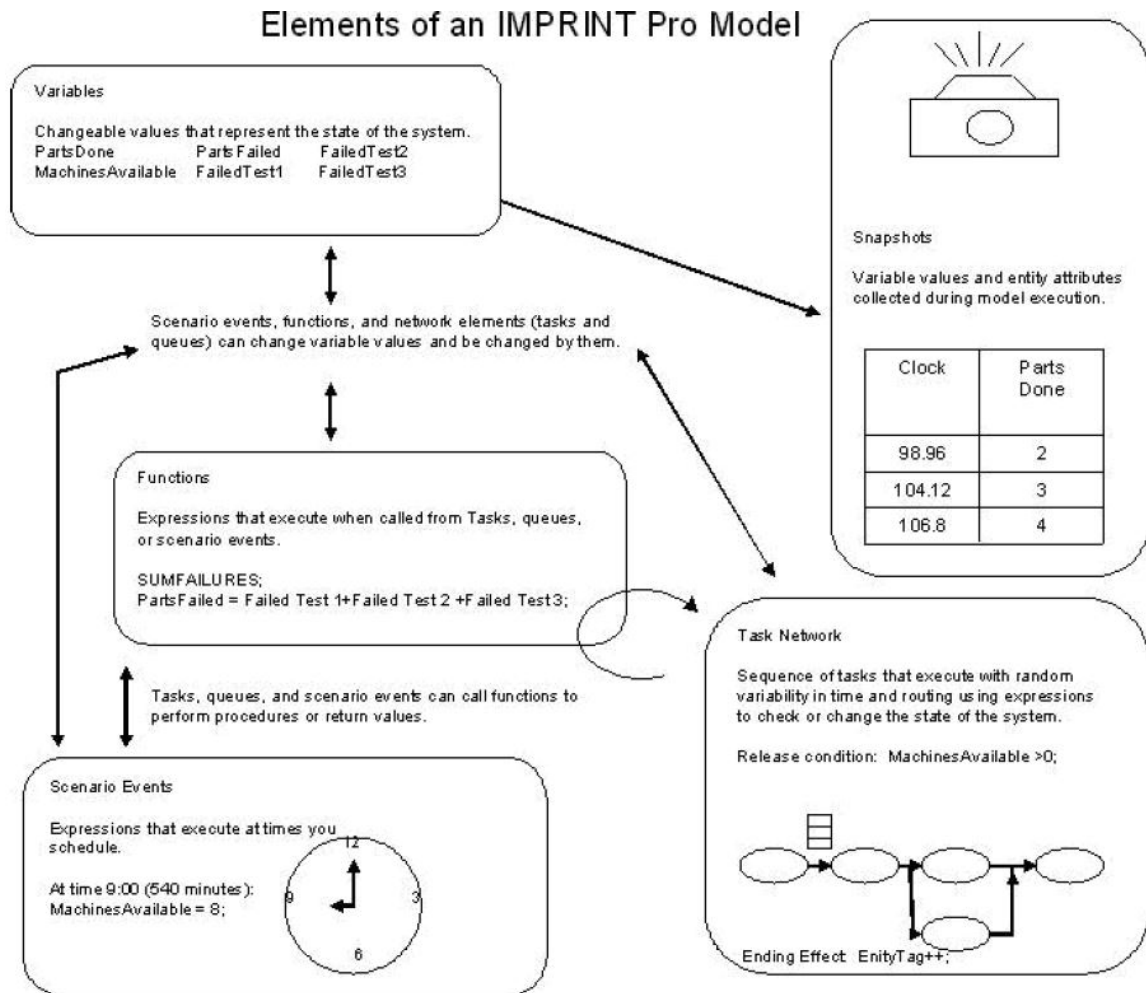


Figure 2.23: Key elements and general flow of an IMPRINT Pro model [44]

workload measurement techniques providing a dynamic, traceable assessment throughout the simulation.

The combination of the workload assessment techniques will satisfy the requirements for this methodology. NASA TLX provides binning of tasks and operational studies which can be used as a baseline. Eye tracking provides the low-level studies to map tasks to workload levels. IMPRINT Pro demonstrates a mapping of tasks to workloads in a simulation environment. Using the information from CSE, specifically from SA, HTA, and workload analysis, then combining it with the methods of OR and System Design will facilitate automation technology trade studies.

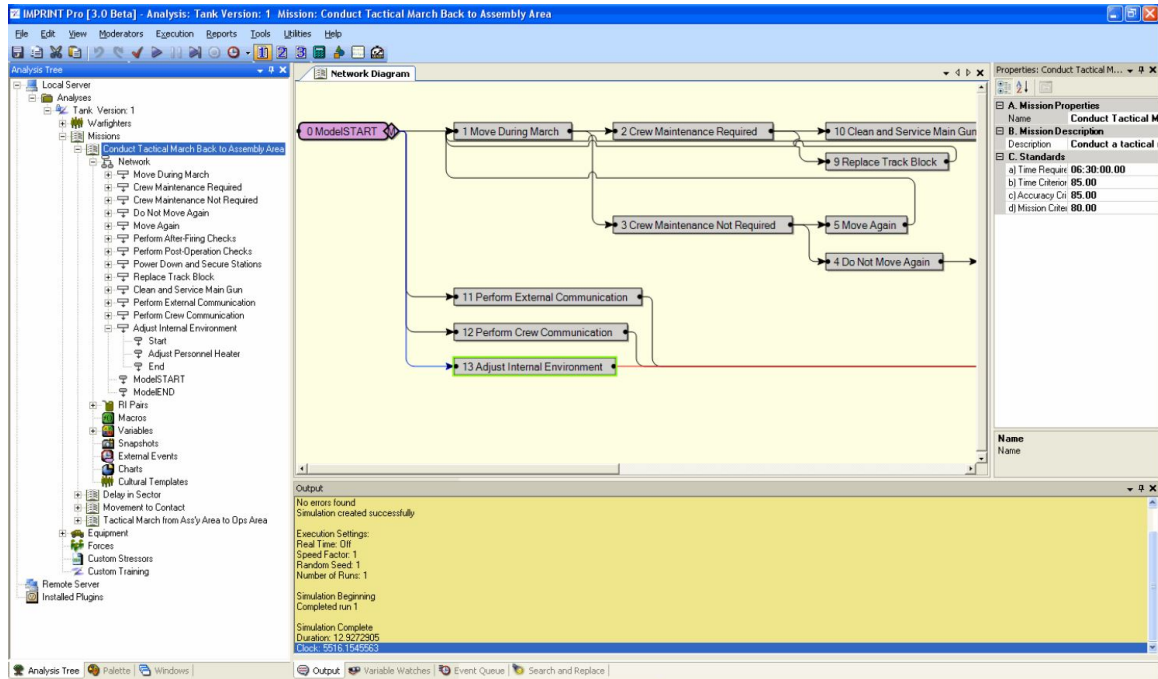


Figure 2.24: Sample network diagram for tasks involved in a tank tactical march back to an assembly area [44]

Maintenance Task Taxon Mappings

This form allows you to modify the mapping of taxons and taxon values for maintenance tasks.

Any changes made here will be saved with the currently selected Analysis.

Maintenance Task	Taxon 1	Taxon 1 Weight	Taxon 2
Adjust & Repair	Fine Motor Discrete	0.50	Visual (Pattern Recognition - Discrimination)
Inspect	Visual (Pattern Recognition - Discrimination)	0.70	Cognitive (Problem Solving and Decision Ma
Remove & Replace	Fine Motor Discrete	0.70	Visual (Pattern Recognition - Discrimination)
Test & Check	Visual (Pattern Recognition - Discrimination)	0.60	Cognitive (Problem Solving and Decision Ma
Trouble Shoot	Cognitive (Problem Solving and Decision Making)	0.80	Visual (Pattern Recognition - Discrimination)
Access	Fine Motor Discrete	0.70	Visual (Pattern Recognition - Discrimination)
Adjust	Fine Motor Discrete	0.60	Visual (Pattern Recognition - Discrimination)
Align	Fine Motor Continuous	0.60	Visual (Pattern Recognition - Discrimination)
Calibrate	Fine Motor Discrete	0.60	Visual (Pattern Recognition - Discrimination)
Camouflage	Gross Motor Light	1.00	
Clean	Gross Motor Light	0.80	Visual (Pattern Recognition - Discrimination)
Debug	Cognitive (Problem Solving and Decision Making)	0.80	Visual (Pattern Recognition - Discrimination)
Disassemble/Assemble	Gross Motor Light	0.50	Cognitive (Problem Solving and Decision Ma
Dispose	Gross Motor Light	1.00	
Evaluate	Cognitive (Problem Solving and Decision Making)	0.80	Visual (Pattern Recognition - Discrimination)
Fault Location	Cognitive (Problem Solving and Decision Making)	0.70	Visual (Pattern Recognition - Discrimination)
Install	Gross Motor Light	0.50	Cognitive (Problem Solving and Decision Ma
Load/Unload	Gross Motor Heavy	1.00	
Lubricate	Fine Motor Discrete	0.70	Visual (Pattern Recognition - Discrimination)

Reload Defaults OK Cancel

Figure 2.25: Mapping between taxons and a maintenance task within IMPRINT [44]

Model Formulation

Before proceeding, a summary of Research Question 1 has been provided in Table 2.7. The first research question focuses on the four pieces required for a methodology to reduce operator uncertainty and conduct automation technology trade space exploration. With all four parts of the model discussed, it is time to identify a suitable agent-based modeling environment. Although a custom solution could be created, this would be a time consuming process so an existing application will be leveraged. There are many agent-based models that have been created. A 2016 study looked at the relative usage of different simulation packages based on studies published at WinterSim 2016 and LinkedIn user groups, shown in Figure 2.26 [39]. Note that some of these software packages, such as Arena, do not specialize in agent-based simulation even if they are capable of workarounds to model agent behaviors. The study also notes that NetLogo has larger user bases on platforms other than LinkedIn. Similarly this study was based on information that would be unlikely to include the Department of Defense's primary tool Advanced Framework for Simulation, Integration and Modeling (AFSIM). Based on this information, four simulation packages have been chosen and evaluated based on their prevalence and/or application to this research: NetLogo, Simio, AnyLogic 7, and AFSIM.

Introducing all four agent-based simulation packages, NetLogo was developed at Northwestern University in 1999. It is heavily utilized by academia because its simplicity and numerous demos quickly allows new developers to create models, however, it has also been used to support research [115]. Its free, open source nature allows it to quickly be utilized by any project. NetLogo is a 2D visualization environment, however, more recently a NetLogo 3D was developed allowing the creation of 3D worlds, although there is less related reference material and body of research using the 3D variant. The traffic demo included in the NetLogo download is shown in Figure 2.28. Simio Simulation Software was introduced to the market in 2008. Simio, like the next two software packages, uses object-based programmings with processes included in the objects as needed[88]. This allows depen-

Table 2.7: A summary of Research Question 1 with its related sub-questions and assertions/hypothesis

RQ	
1.0	Can the inclusion of cognitive engineering disciplines in the operational assessment of systems reduce the uncertainty pertaining to the operator's cognitive load by providing a dynamic, traceable, and quantifiable measurement throughout the scenario?
1.1	Can a dynamic task list be created from the operator's perspective to map operator awareness to a goal-action agent-based rule set?
	Assertion: utilizing HTA and representing goal-action sequences will enable the creation of a dynamic task list that can create an agent rule-set capable of guiding behaviors based on operator awareness
1.2	Can the operator's knowledge acquisition and information utilization processes be explicitly modeled to include acquisition delays and utilization challenges which directly impacts awareness and how the agent follows its dynamic rule set?
	Assertion: the studies conducted in the field of SA will allow for the dynamic representation of the costs of information acquisition and the challenges with information utilization which can be utilized in the activation sequence of the dynamic task list
1.3	Can the typically manual, human-based workload measurement techniques be applied in an agent-based environment to provide a quantitative assessment of workload?
	Hypothesis: the detailed task list will map to different elements of the workload measurement techniques providing a dynamic, traceable assessment throughout the simulation

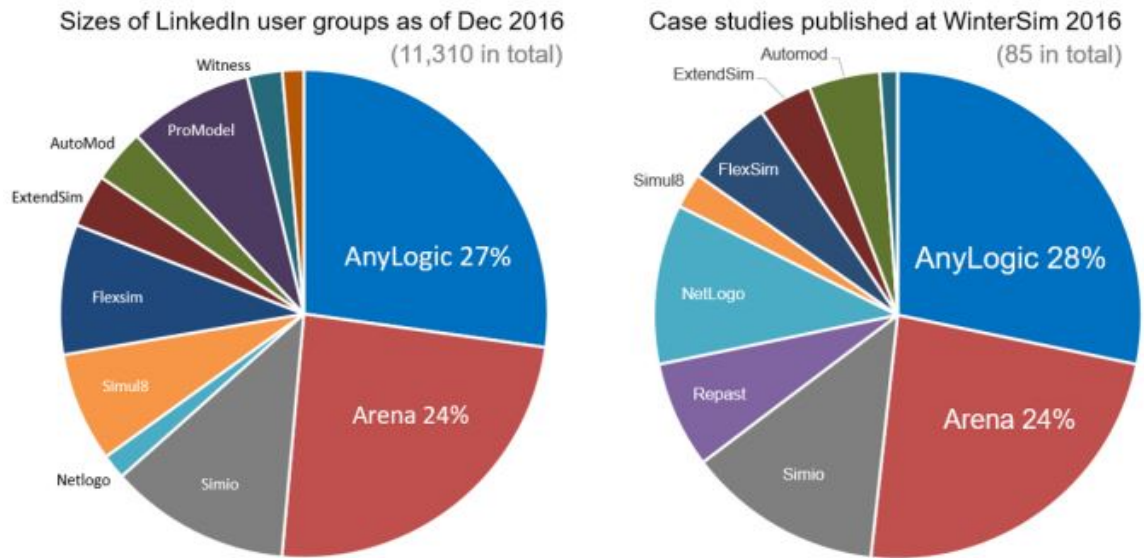


Figure 2.26: Simulation tools approximate market shares based 2016 study on LinkedIn user groups and publications at WinterSim 2016 [39]



Figure 2.27: Example of a Simio agent-based model and visualization utilized in the modeling of an airport terminal sales [2]

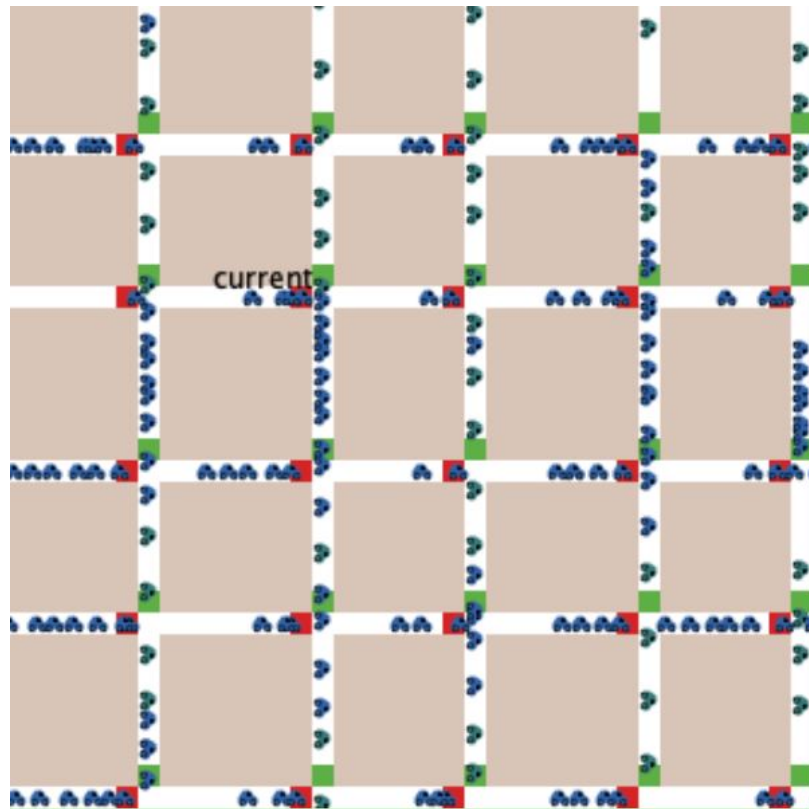


Figure 2.28: Example of a NetLogo agent-based model and visualization utilized in the modeling of a traffic grid [115]

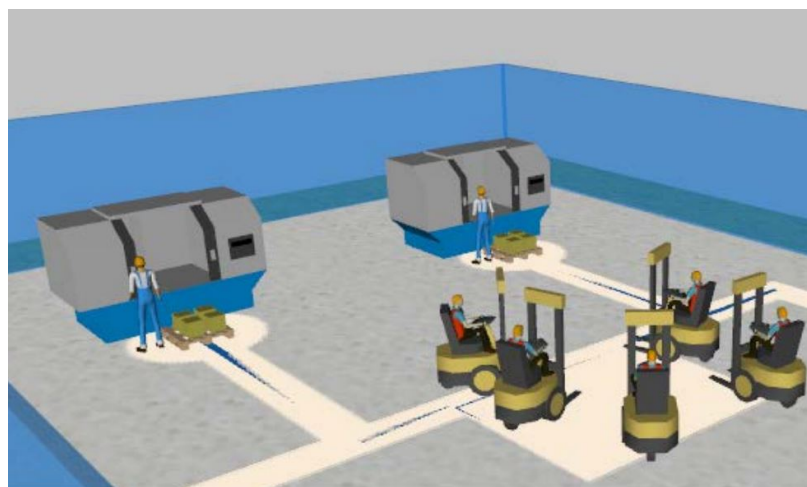


Figure 2.29: Example of an AnyLogic agent-based model and visualization showing fork lift movement to load machinery in a factory model [38]

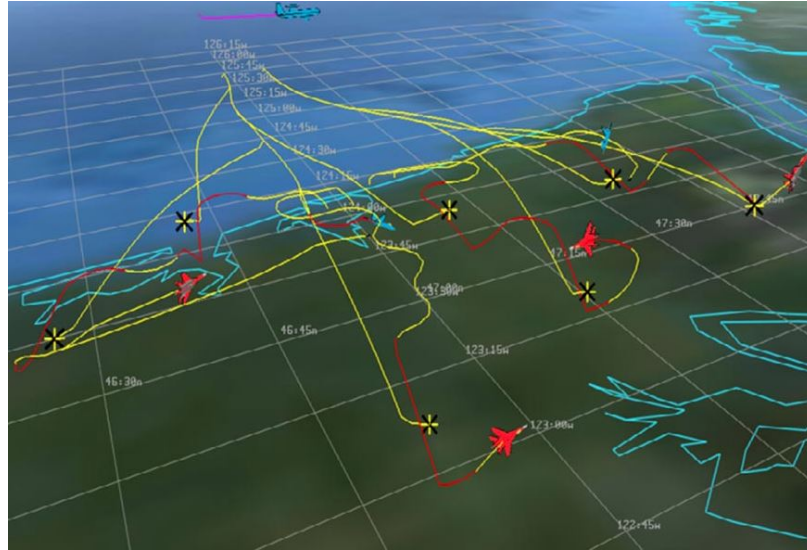


Figure 2.30: Example of an AFSIM agent-based model and visualization depicting an air combat scenario [111]

dencies to be captured: an agent is not allowed to do an action until the process allows it or the process may not proceed until an agent is free. An example Simio model and corresponding visualization for an airport terminal is shown in Figure 2.27. AnyLogic was created in 2000 by the AnyLogic Company with AnyLogic 7 being released in 2014. It supports agent-based, discrete event, and system dynamics modeling, to allow the creation of composite models for more detailed modeling [38]. A process-based AnyLogic model depicting the loading process for a machining center is shown in Figure 2.29. The last one being compared is AFSIM. It was originally developed by Boeing starting in 2005, but transitioned ownership in 2013 and it is continually updated by the US Air Force [17]. AFSIM has become a standard simulation framework within the Department of Defense (DoD) and among DoD contractors. Its growing library of models (primarily focused on military applications) and class-based architecture allow quick agent and environment creation, while its open framework allows missing components or higher-fidelity requirements to be satisfied as needed[17]. An air combat example from the Infoscitex training website is shown in Figure 2.30. With all four simulation packages identified, the next step is choosing an appropriate agent-based model.

The metrics by which the software packages were evaluated are: detailed environment, agent library, stochasticity, pre-built sensors which can be customized, physics-based movement, custom behavior trees, custom task output, flow diagrams, visual output, open-source, and freely available. These were chosen based on the complexity of model creation, needs of the task, sensory, workload, and operations models, and the overall requirements of the methodology. The detailed environment pertains to the number of elements and simplicity of adding elements to the environment which will be perceived by the software agent. The framework must allow sensor creation and provide physics-based model representation to accurately portray the sub-systems and system performance. Behavior trees and task output are critical for the dynamic task list and workload model respectively. Flow diagrams and visual output aid in traceability, coding, and debugging but are not required. Freely available must be satisfied due to limited resources, and open-source ensures any limitations in the agent-based framework may be overcome when trying to integrate additional functionality (however this is a soft requirement, because some simulation environments may have all the functionality required). An overview of these criteria in relation to each software package is shown in Figure 2.31. Based on this analysis AFSIM was chosen, which is freely available to DoD contractors. Primary drivers for the selection was its availability within the lab and its flexibility as an architecture. It also provides many basic models for DoD applications, which is a domain of interest for this research. Lastly, AFSIMs extensive and growing utilization within the DoD provides greater applicability of this methodology to existing technology trade studies. AnyLogic does provide better applicability to commercial applications based on its existing libraries, and its interface could provide further information with flow diagrams. However, it does not seem capable of capturing the fidelity within sensors or provide extensive physics models, and only the academic version was available which did not satisfy the needs of this research.

Based on the above discussion on modeling components and the selection of an agent-based framework, the hypothesis for Research Question 1.0 is:

Features	NetLogo	AFSIM	AnyLogic	Simio
Detailed environment	No	Yes	Yes	Yes
Agent library	Demos	Military	Commercial	Commercial
Stochasticity	Yes	Yes	Yes	Yes
Pre-built, customizable sensors	No	Yes	No	No
Physics-based movement	No	Yes	No	No
Custom Behavior Trees	No	Yes	Yes	Yes
Custom Task Output	No	Yes	Unknown	Unknown
Flow diagrams	No	No	Yes	Unknown
Visual output	Yes	Yes	Yes	Yes
Open-source	Yes	Yes*	No	No
Freely Available	Yes	Yes*	Academic	Academic

* Distribution free to Department of Defense and contractors

Figure 2.31: Matrix of alternatives describing the four chosen agent-based simulation packages with evaluation metrics [38, 115, 17, 82]

Hypothesis 1.0: The inclusion of a dynamic task list, SA model, and workload model in an agent-based operations model (AFSIM) will enable a dynamic, traceable, and quantifiable operator workload measurement similar to those currently manually conducted.

The integration of the four modeling components (operations model, dynamic task list, sensory/SA model, and workload model) will move the operator's workload from a static input to a dynamic output. This dynamic output will vary based on the environmental factors such as density of obstacles and opposing agents, and provide a metric for requirement definition and technology evaluation. The integration of these components in a modeling environment allows driving factors of workload to be directly traced, and possibly mitigated during the early development process.

2.4.3 Model Solution

The first experiment needs to test the hypothesis for Research Question 1.0 and the dependent Hypothesis 1.3. This can be done through a comparison between the results and general trends of an agent-based model to a study using the NASA TLX scale and participants. This requires the modeling of a system that has been well studied. The system and environment to be modeled must have performance data, environmental data, operator task lists, operator awareness times, and workload studies which can be used to evaluate the dynamic output provided by the agent-based model. Due to the limited publicly available data for automation on aviation systems, the automotive field is again utilized. The data requirements for creating an operator-focused agent, benchmarking the workload data, and capturing then assessing different automation technologies drove the decision to test the methodology for capturing operator workload during a driving scenario under multiple traffic conditions. Traffic and the act of driving are well studied. Vehicle performance characteristics are easy to find and capture in a physics-based mover. The 2-dimensional domain aids in the initial test case for operator decision making and environment capture. The environment is well defined but variant based on the time of day and location, which can provide stressors to the dynamic workload assessment. The study by the NHTSA [62] provides the information to define a dynamic behavior tree for one of the primary factors in the driving process, merging. This detailed behavior and experience in the task of driving will provide adequate information to define additional software agent behaviors. The NHTSA study also provides information regarding the perception processes of a driver, such as the time taken to acquire different awareness elements. Lastly, there is a presence of studies which can be used to benchmark this new methodology, such as the work of Rahman et. al which used the NASA TLX to subjectively study real-time driver workload in three different environments of different complexity [86].

The setup for the experiment results in the creation of an agent-based model using AFSIM that is capable of assessing the cognitive load associated with driving in different

two scenarios (city and highway). The software agent's internal domains are improved based on the three operator focused model components: dynamic task list, SA model, and workload model. The dynamic task list and SA model are built out to represent the general driving activity using the data from NHTSA and similar studies. The environment model captures three different driving complexities: simple (highway) and complex (city). The basics of these situations are taken from Rahman et. al's study and are shown in Table 2.8 [86]. Road networks are created within the AFSIM model based on existing infrastructure in Georgia and standard rules of the road are used to specify external constraints on the software agent, similar to those experienced by the operators in the comparative study. The goal of the operator in the simulation is to traverse between randomly assigned start and end points. Throughout the driving activity, workload is output to test the functionality of the methodology and enable correlations between measurements and variations to the environment. Once the basic mental model of tasks and awareness are created, it will not be changed between environment variations, one mental model must be able to capture the workload differences from the environment.

The output of this experiment is a dynamic measurement of the operator's workload in different environmental conditions. The workload is measured using the NASA TLX scales, providing measurements for mental demand and temporal demand. While the performance is monitored based on time taken to reach the destination compared with the ideal time to reach the destination. These three metrics will be compared to the results of Rahman et. al study to look for similar trends and insights. The results from Rahman et. al's study are shown in Figure 2.32. All three measurement metrics should show differences through changes to the environment. The operator workload was found to be the highest in the rural case due to the uncertainty of obstacles and the higher rate of travel. Although quantitatively the agent-based model will not match the results here, unless a quantitative baseline can be determined, the trends should remain similar. If successful and the agent-based model is capable of providing a quantitative analysis of workload then Hypotheses

Table 2.8: Environment settings for the agent-based model. From Rahman et. al’s human-in-the-loop study [86].

	Simple Situation	Complex Situation
Road design	Highway	City Road
Speed limits	110km/h (68 MPH)	60 (37 MPH) to 90km/h (56 MPH)
Road layout	Straight	Many curves
Traffic flow	Low density	High density

Table 2.9: A summary of Experiment 1.0 which aims to test the hypothesis and sub-hypothesis of Research Question 1.0

Objective	Compare the results and general trends of an agent-based model to a study using the NASA TLX scale and participants
Setup	Create an agent-based model using AFSIM to assess the cognitive load associated with driving in different scenarios - The dynamic task list and SA model will be built out to represent the driving activity - The goal of the operator (driver) will be traversing between randomly assigned start and end points
Output	Workload variations across the 3 TLX (mental demand, temporal demand, performance) scales based on changes to scenario (highway, city, rural) but not varying the agent’s mental model

1.0 and 1.3 will be shown to be true and technology integration can start. A summary of Experiment 1 is shown in Table 2.9.

2.4.4 Technology Trade Space Evaluation

Research Question 1 focused on the quantification of workload, which also tackles Gaps 1 through 3. Research Question 2 focuses on the assessment of different automation technologies. In order to do this, technology must be captured within the agent-based model. Research Question 2.0 is formally stated as:

Research Question 2.0: *How should variations in the system due to technology integration be captured within a modeling environment to properly capture the*

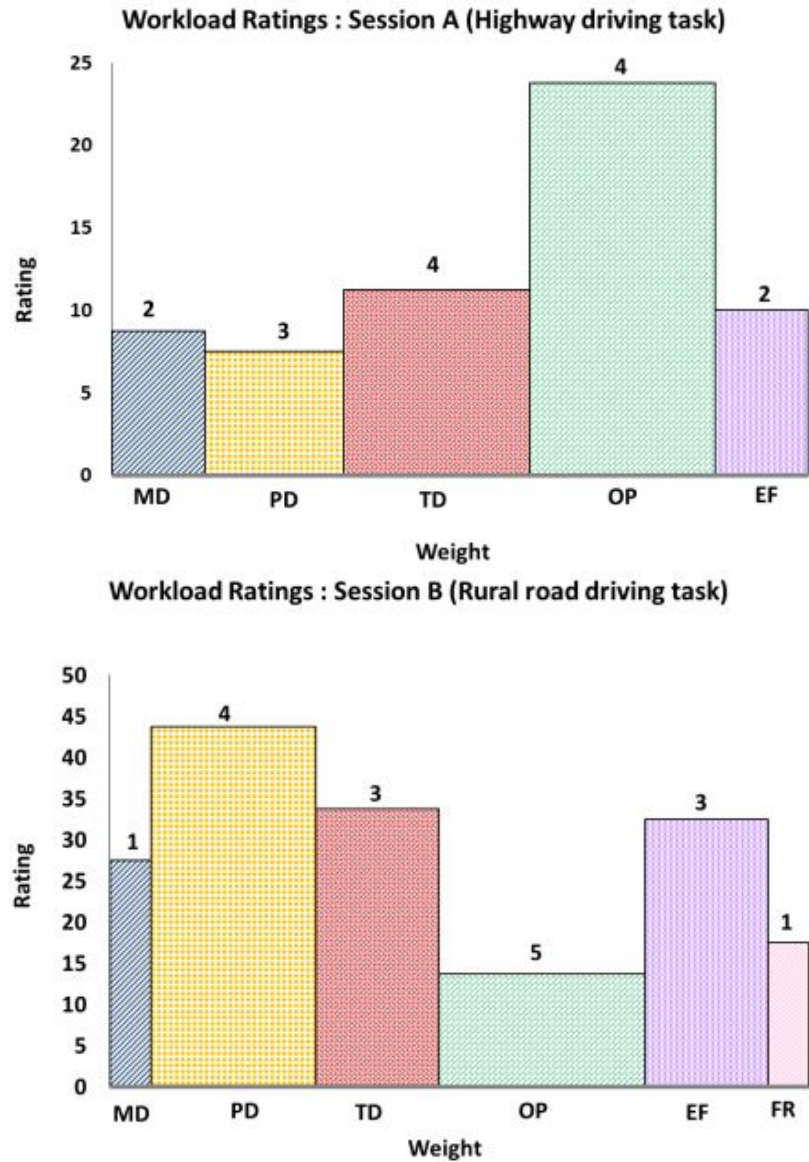


Figure 2.32: NASA TLX workload ratings from a study assessing driver workload under different environmental conditions (highway [simple], rural [moderate], and city [complex]) [86]

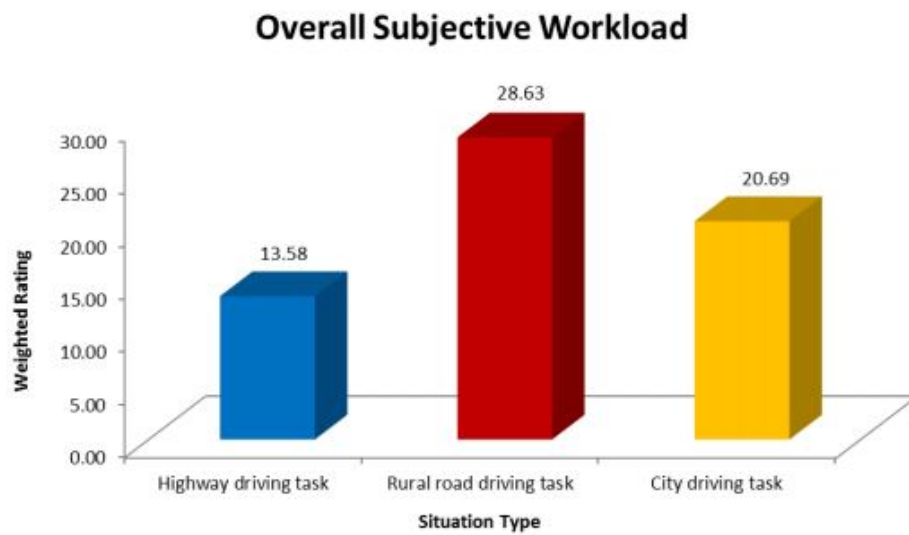
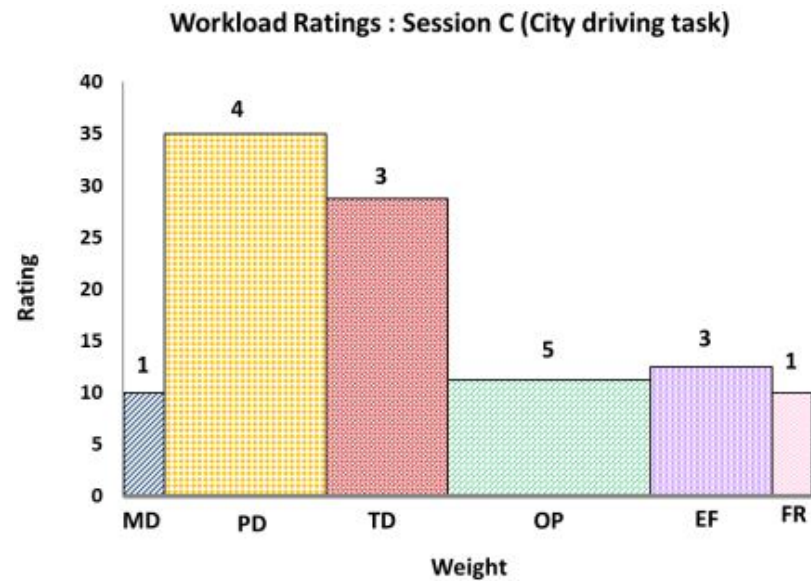


Figure 2.32: NASA TLX workload ratings from a study assessing driver workload under different environmental conditions (highway [simple], rural [moderate], and city [complex]) (cont.) [86]

effect on the system's performance and enable the assessment of changes to the operator's cognitive load?

Based on the gap discussion, these changes take two forms. The first form (Gap 4) is changes to the task list and behavior tree mapping based on the implementation of the automation technology. The addition of an automation technology changes the methods available and how the operator may perform a goal, so it adds or modifies branches on the behavior tree. The second change pertains to the level of trust that the operator has with the automation (Gap 5). This trust determines the frequency a management branch may be visited thus has a direct effect on the awareness of the operator, the operator's workload, and the resulting performance of the system. Each of these will be discussed in more detail below with a respective research question.

Variations to the Mental Model

The first issue to be addressed regards capturing changes to the quantified operator workload based on the inclusion of automation technology. "Embracing a new era where 'joint warfare' means human-robot teams requires a better understanding of autonomy and a better effort to design for human-machine interdependence" [110]. The agent-based model must be capable of capturing the difference in workload and performance for systems utilizing different types and levels of automation. These changes occur in the mental model (dynamic task list), and drive changes to measured workload. Stated formally Research Question 2.1 is:

Research Question 2.1: *What changes must be done to the agent's mental model to capture changes in technology or automation of the system?*

The internal domains of the software agent have already been defined using detailed behaviors trees, based on operator tasking. However, now these trees must be modified based on the technology being captured. The inclusion of technology modifies the operator's role

from sole-actor to additional functions in management and intervention. In example, Figure 2.20 shows the operation event sequence diagram of changing lanes during a normal driving activity, however many of these actions are handled by an automation technology like Tesla's Autopilot [109]. The operator's role during highway driving is simply automation monitoring and intervention. Therefore, the inclusion of automation technology adds two additional branches to the standard behavior tree: monitoring and correction. The monitoring branch describes the tasks taken by the operator to enable, disable, and utilize the automation technology. This branch represents the operator's awareness of the automation system and will be modified based on the trust level. The correction branch specifies the set of activation criteria by which the operator will understand the system is not functioning as desired and take over. The correction branch dictates which actions the operator will take to manage the technology. This leads to Assertion 2.1:

Assertion 2.1: Adding monitoring and correction branches to the dynamic task list will modify the agent's behavior and task load differing amounts based on the technology integrated.

Operator Trust

With the dynamic task list and resulting behavior tree adjusted to include new branches for technology, it is also important to understand the sensitivity of the automation technology to operator trust. As more automation is integrated, there is an implicit trust which has led to many errors by operators with differing experience levels. A novice (or trusting operator) will fully utilize the automation, while an expert (someone who understands the edge cases) will constantly double check the automation results if they can. This was discussed in the motivations section with the NUWC study and is shown in Figure 1.5 [56]. Therefore, Research Question 2.2 is:

Research Question 2.2: How should the operator's trust be captured within the

simulation such that the Situational Awareness and task loading of the operator are adjusted based on the level of trust in the automation?

One of the most significant drivers of workload is the operator's trust in the system. Experts will use equal amounts of automation computations and individual measurements; ensuring agreement and proper functionality. Novices overly rely on the system's intended functionality, rarely double-checking automated calculations and actions. This double checking of data results in higher workload but also can result in higher system performance due to increased SA. The dependence between awareness and performance was shown in the ATC technology utilization study by Edwards et. al [27]. It is important to capture this dependence in an agent-based model when assessing an operator's automation management role. Higher knowledge overlap between the vehicle and the operator results in higher workload, but it also creates a shared awareness to maintain desired levels of performance. It is proposed that this level of trust can be captured by varying the frequency certain SA related tasks are conducted:

Hypothesis 2.2: By changing the duration which data is stored in the operator's SA model (if also monitored by the automation) and the frequency of monitoring tasks based on the level of trust in the technology, it will approximate the performance differences shown by different skill-level operators.

Experiment

With the two elements of Research Question 2.0 discussed, the overall hypothesis can be introduced:

Hypothesis 2.0: The modifications, additional two branches on the dynamic task list and variations to SA according the operator trust, will allow technology trade space exploration based on operator workload and system performance.

Table 2.10: A summary of Research Question 2 with its related sub-questions and assertions/hypothesis

RQ	
2.0	<p>How should variations in the system due to technology integration be captured within a modeling environment to properly capture the effect on the system's performance and enable the assessment of changes to the operator's cognitive load?</p> <p>Hypothesis: The modifications, additional two branches on the dynamic task list and variations to SA according the operator trust, will allow technology trade space exploration based on operator workload and system performance.</p>
2.1	<p>What changes must be done to the agent's mental model to capture changes in technology or automation of the system?</p> <p>Assertion: Adding monitoring and correction branches to the dynamic task list will modify the agent's behavior and task load differing amounts based on the technology integrated.</p>
2.2	<p>How should the operator's trust be captured within the simulation such that the Situational Awareness and task loading of the operator are adjusted based on the level of trust in the automation?</p> <p>Hypothesis: By changing the duration which data is stored in the operator's SA model (if also monitored by the automation) and the frequency of monitoring tasks based on the level of trust in the technology, it will approximate the performance differences shown by different skill-level operators.</p>

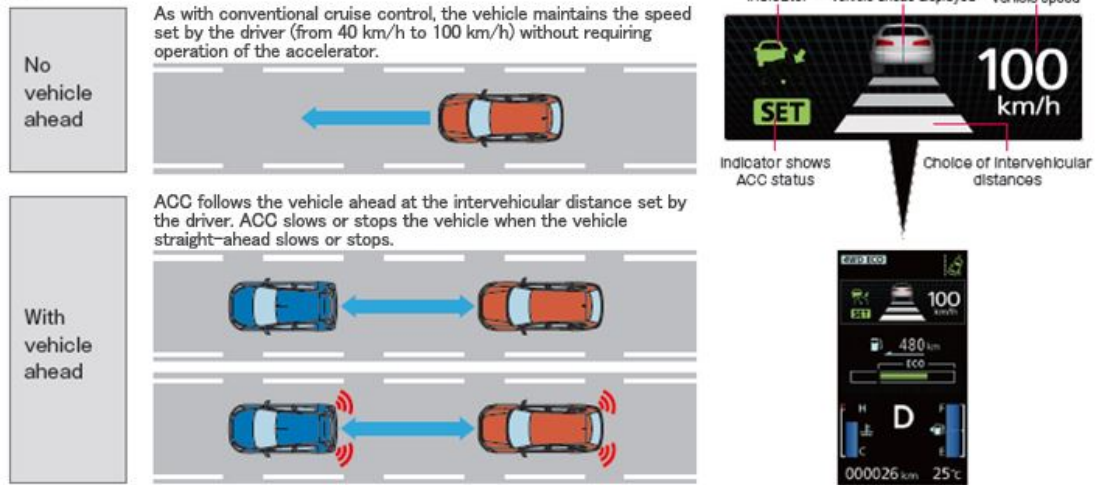
Research Question 2 is focused on the development of the technology trade space. A summary of all components of Research Question 2.0 is shown in Table 2.10. The easiest way to test this hypothesis is through the expansion of the agent-based model from Experiment 1, since the software agent has already been benchmarked for workload quantification. Experiment 2 tests whether the integration of different technologies can be shown through changes in calculated workload and variations in agent performance.

The setup for Experiment 2 will be modifications to the sensor components and mental model of the software agent used in Experiment 1. The environment, goal, and test

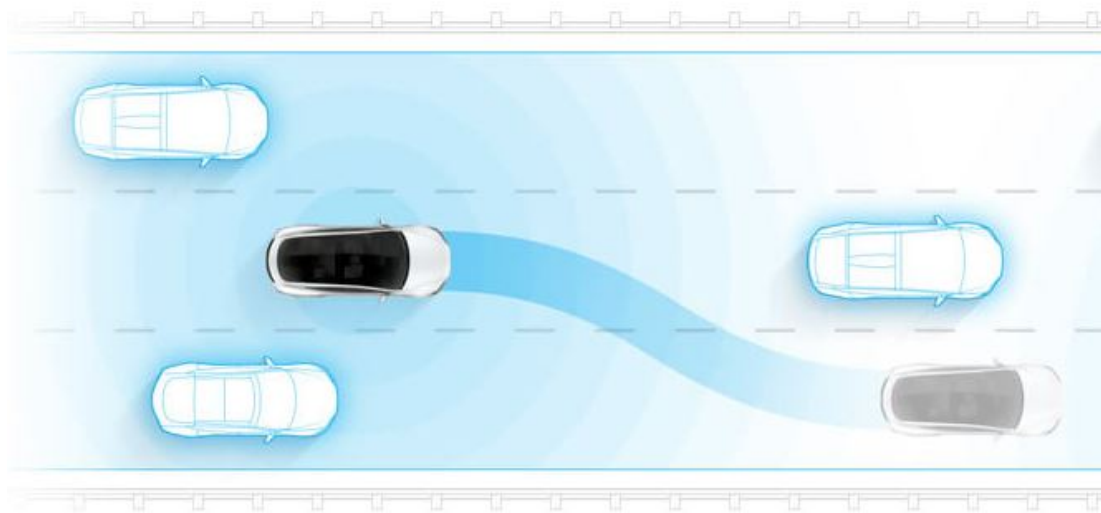
cases will remain the same. The assessed technologies will be: adaptive cruise control and highway autopilot. Adaptive cruise control (ACC) reduces the speed of the subject vehicle as it approaches another vehicle to maintain a safe following distance, when more inter-vehicular distance is achieved then the car will accelerate back to the desired speed, shown in Figure 2.33a. The technology represents automation in the longitudinal direction, while the operator is responsible for lateral corrections. The operator overtakes for the ACC when the speed is below a desired threshold by changing lanes. The second modeled technology, highway autopilot, is shown in Figure 2.33b. The blue around the car represents the automation's awareness through cameras and ultra sonic sensors. The autopilot controls both the lateral and longitudinal travel directions, and is shown changing a lane to avoid a slower car. This automation technology moves the operator to purely a manager role, but since the operator is managing the technology they are assumed to be maintaining an awareness similar to the autopilot. Technologies will be added independently to the vehicle. Each technology integration will change the steps and responsibilities of the driver as they perform the goal of traversing between two random points. This requires the software agents mental model to be changed. The task sequencing will change based on the integrated technology; the behavior tree will gain two new branches as discussed above. And the operator trust level must be specified as an input to determine the level of knowledge overlap between the vehicle and the driver: are both keeping track of all vehicles or will the driver only respond when alerted?

Experiment 2 will develop an initial technology trade space with performance and workload metrics on the system. The trade space will have the vehicle without technology and with all combinations of the above technologies. Comparisons and trends will be analyzed to compare system effectiveness with workload, and the results from the agent-based model will be compared to workload assessment by the automotive industry on the effects of automation integration (primarily they focus on performance). A summary of Experiment 2 is shown in Table 2.11.

[System Operating Patterns]



(a) Adaptive cruise control (ACC). Source: Mitsubishi Motors [26]



(b) Highway autopilot (Autopilot). Source: Tesla [109]

Figure 2.33: Overview of functionality of driving automation technologies integrated in the agent-based model. Figures from:

Table 2.11: A summary of Experiment 2.0 which aims to test the hypothesis and sub-hypothesis of Research Question 2.0

Objective	Compare the dynamically calculated workload and performance changes after automation integration with studies done for automated cars
Setup	Modify agent-based model from Experiment 1 Add technologies: - Forward collision avoidance - Lane departure warning - Adaptive cruise control Modify the mental model: - Task sequencing must be changed to include variations introduced by the technology - Operator trust level must be specified as an input to determine level of knowledge overlap and workload
Output	Compare cognitive workload changes with those of studies done for automated cars

2.4.5 Interpretation

Measurement of workload and the development of a trade space is only part of the methodology, the last step of the OR process is interpretation which requires output analysis. This step is focused on solving gap 6, presentation of findings to a decision maker. By using the new data collection on workload, new insights are provided through a reduction in operator uncertainty by which, it is proposed, that better technology decisions may be made. This leads to the final research question for benchmarking the methodology:

Research Question 3.0: What effect will reducing the uncertainty around the operator's cognitive load, by enabling the quantification and identification of the cognitive factors, have on performance evaluations and technology integration decisions?

The workload quantification within the agent-based model allows operator workload to be graphed over time and tie periods of high/low demand to environmental and system

factors. A representative graph is shown in Figure 2.34, from an article comparing cognitive load for graphical and voice user interfaces [114]. This graph shows how a difference in technology can be used to drive desirable operator workload. The voice interface has higher peak moments, however, the valleys may be undesirable. Whereas, a graphical interface does not garner nearly the attention of the operator. This could represent one subsystem in the model and may be easy to comprehend. However, automation technology's dependence on the environment and the combination of sub systems makes the creation of notional graphs more challenging. Of similar interest to a decision maker is the operator's awareness and its dependence on performance. Going back to the Tesla crash discussed in the motivation, the system failure (crash) was blamed on multiple issues: "his inattention and overreliance on the car's advanced driver assistance system; the Tesla's "Autopilot" design which permitted the driver to disengage from the driving task; and the driver's use of the system in ways inconsistent with guidance and warnings from Tesla" [109]. Note that system design and operator awareness are both covered as causes. The operator had a low level of workload for many instances of the trip, shown in Figure 2.35. The driver was allowed to disengage from the driving task for 20 minutes preceding the crash. As discussed and shown by the ATC automation study, operators who are too disengaged from the task typically do not maintain a sufficient level of awareness to sustain desired levels of performance. This is especially true for novice operators, who have high levels of trust in the system, because they are less likely to maintain overlapping awareness. Therefore it is hypothesized that this new methodology is required to identify similar operator workload and awareness characteristics to enable more informed decisions for partially automated systems.

Hypothesis 3.0: Reducing uncertainty concerning cognitive load will allow decision makers to locate possible failure points due to: peaks, valleys, or overall high task loading, or inadequate situational awareness.

Peaks of Attention: Graphical vs Voice User Interfaces

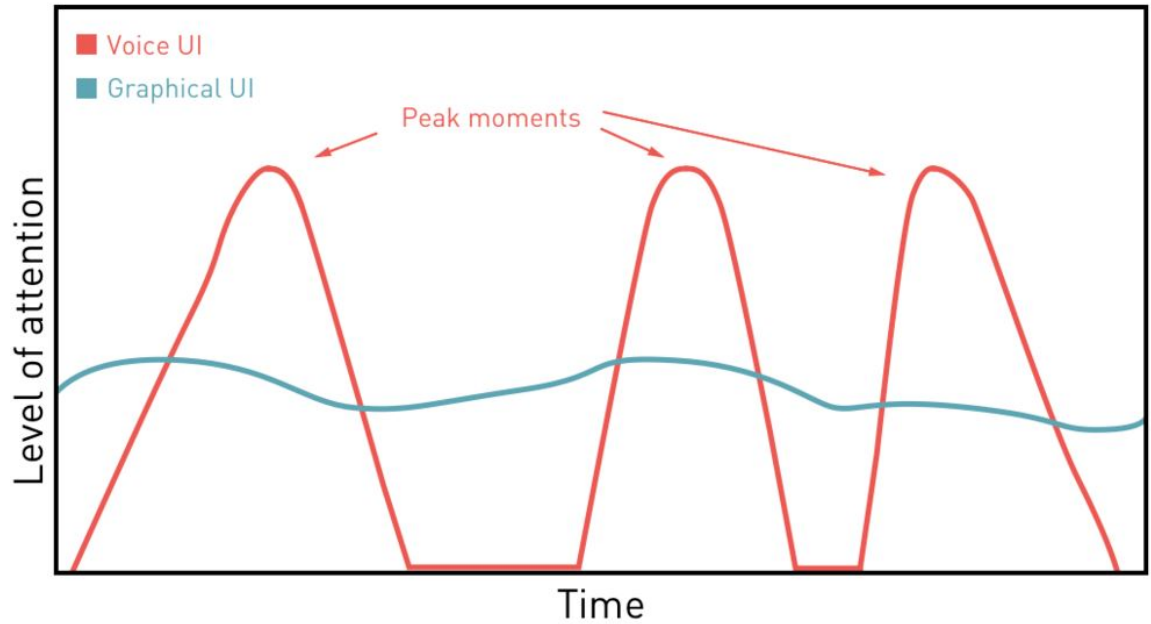


Figure 2.34: Notional graph showing cognitive workload over time for a graphical versus voice-based user interface [114]

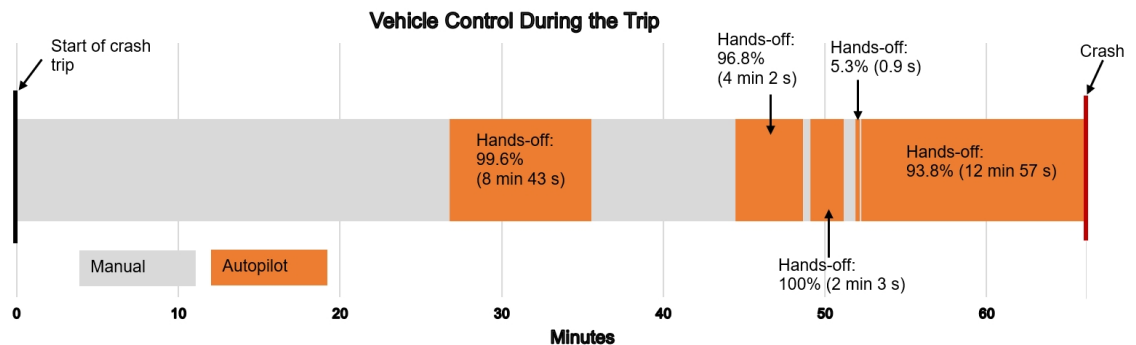


Figure 2.35: NTSB graph showing the amount of operator control before the 2018 Tesla crash [109].

Table 2.12: A summary of Research Question 3

RQ	
3.0	<p>What effect will reducing the uncertainty around the operator’s cognitive load, by enabling the quantification and identification of the cognitive factors, have on performance evaluations and technology integration decisions?</p> <p>Hypothesis: Reducing uncertainty concerning cognitive load will allow decision makers to locate possible failure points due to: peaks, valleys, or overall high task loading, or inadequate situational awareness.</p>

Experiment 3

Experiment 3 is focused on output analysis and testing Hypothesis 3.0; a summary of Research Question 3.0 is shown in Table 2.12. The objective of this experiment is to assess the capabilities of the operator workload measurement framework. The results from Experiment 2 will be used to create an interface that allows decision makers to understand operator workload throughout the mission assessment. A similar output will be created to graphically represent the operator’s awareness. These figures will be used to create comparisons between technology combinations in the trade space. The comparisons show how each technology effects the three NASA TLX workload categories (Mental Demand, Temporal Demand, and Performance) and the operator’s awareness over time based on trust level. The goal of the experiment is to compare this dynamic workload model results with the NTSB Tesla study (shown in Figure 2.35) and similar studies, to compare predicted driver attention and workload with actual numbers. This new methodology does not focus on defining hard limits for either awareness or workload, it merely reduces the uncertainty surrounding both so that the decision maker may better select a technology set. A summary of Experiment 3 is shown in Table 2.13.

Table 2.13: A summary of Experiment 3.0 which aims to test the hypothesis of Research Question 3.0

Objective	Assess the capabilities of the operator workload measurement framework
Setup	Analyze the simulation outputs from Experiment 2: - Create an interface for decision makers to understand operator workload throughout the mission - Graph the data being kept in the Operator’s awareness model throughout the simulation. - Enable the comparison of different system technologies
Output	Compare the outputs of the dynamic driving assessment to the Tesla studies on driver attention and workload prior to accidents

2.4.6 Demonstration

The last step of this work is testing the overall methodology on a use case to ensure that the research objective has been achieved and the methodology can be applied to early-design aerospace vehicles. This leads to Research Question 4.0 which states:

Research Question 4.0: Can the operator-focused methodology provide insights to a current area of focus for the aerospace domain through the explicit modeling of the piloted aircraft and the automation technology?

Due to the growing interest within the DoD and the recent first flight of Boeing’ Airpower Teaming System, the application focuses on an manned-unmanned teaming application. Manned-unmmanned teaming pairs a piloted aircraft with one or more unmanned combat aerial vehicles (UCAV); the pilot is simultaneously responsible his aircraft and for the UCAV’s flight path and payload operation, there is no UAV pilot involved [119]. The perceived pros of this control structure is better satisfaction of the pilot’s need and reduced manpower requirements, however, it also requires pilot cross-training and increased multi-tasking [119]. Based on the expected benefits and challenges of MUM-T, Hypothesis 4.0 states:

Hypothesis 4.0: The operator-focused methodology is capable of decreasing uncertainty in the analysis of manned-unmanned teaming.

Experiment 4.0 uses a basic intelligence, surveillance, reconnaissance (ISR) mission through contested airspace to assess the methodologies capability to assess MUM-T applications. The mission type is chosen because it is expected to be a primary function for manned-unmanned teaming [79]. The implementation compares the performance of the ISR mission by either a solo, piloted aircraft or a MUM-T utilizing a single UCAV. The pilot will have sole control of the UCAV. The experiment focuses on the mission segment between leaving the rally point to conduct the surveillance and returning to the rally point. The ground operations are assumed to be handled by a third party, which is a slightly lower level of MUM-T automation and a good starting point [119]. The initial ISR mission is a piloted aircraft flying through surveillance checkpoints while avoiding enemy radar sites and engaging red air forces when they are detected. This mission layout contains navigation tasks which can be passed to a UCAV and a dynamic element in the air-to-air engagement which is continually handled by the pilot. Once an initial baseline is created for the mission without technology, the UCAV is added to assist the individually piloted aircraft. The UCAV is capable of navigation through the area, however, each time it reaches a checkpoint the pilot must approve its location before it is allowed to proceed. When the UCAV is introduced, it conducts the surveillance segment, while the pilot maintains a combat air patrol. To ensure communication between the pilot and the UCAV, they must maintain a distance within line-of-sight. The differences between the solo aircraft and the team are captured in the dynamic task list and the sensory model for the pilot. ISR tasks are moved to the UCAV, while monitoring and correction branches are added to the pilot's task list. Each case will output the NASA TLX workloads for mental demand and temporal demand, performance, and awareness to create an environment where technology trade-offs may be understood by a decision maker. A summary of this experiment is shown in Table 2.14.

Table 2.14: A summary of Experiment 4.0 which aims to test Hypothesis 4.0 and validate the overall research objective

Objective	Test the methodology on a use case that is of growing interest and a primary motivation of this work: manned-unmanned teaming
Setup	Create an agent-based model for an intelligence, surveillance, reconnaissance (ISR) mission: Capture workload assessment scores throughout the mission: <ul style="list-style-type: none"> - Mental Demand - Temporal Demand - Performance Integrate technology : Unmanned Wingmen <ul style="list-style-type: none"> - Change the dynamic task list and sensory model Build a technology trade space with pilot cognitive factors as one of the metrics
Output	Workload measurements throughout the mission duration across the 3 workload scales A trade space showing the effects of manned/unmanned teaming

CHAPTER 3

BENCHMARKING METHODOLOGY USING AUTOMOTIVE EXAMPLE

To test this new workload methodology, the experiments outlined above need to be conducted and the results analyzed. A summary of the experiments, the hypothesis being tested by each experiment, the research questions, and the overall research objective is shown in Figure 3.1. Experiment 1 focuses on initial validation of the methodology by benchmarking the workload metric. This is done by creating an initial driving environment and comparing the results to existing human-based experimental data. This experiment assesses the capability to differentiate operator workload based on environmental changes. Experiment 2 takes the workload metric further by utilizing the driving environment to assess different automation technologies. This experiment shows how automation affects system performance and the driver's workload. Experiment 3 utilizes the results from Experiments 1 and 2 to provide operational insights to a decision maker. This experiment utilizes the workload and task frequency data to understand and compare how the operator interacts within their environment while utilizing different levels of automation driving technologies. Lastly, Experiment 4 applies this same workload methodology to a manned-unmanned fighter aircraft teaming scenario. This scenario demonstrates the applicability of the method to the aerospace domain, however, it is more reliant on assumptions than the driving experiments due to limited availability of air combat data.

3.1 Experiment 1 - Benchmarking

Experiment 1 tests the hypotheses from Research Questions 1.0 and 1.3 while validating the assertions made in Research Questions 1.1 and 1.2, which can all found in Table 2.7. The overall goal of this experiment is to assess the validity of the proposed workload metric by comparing the simulated results to a human-based study. Data from an existing study

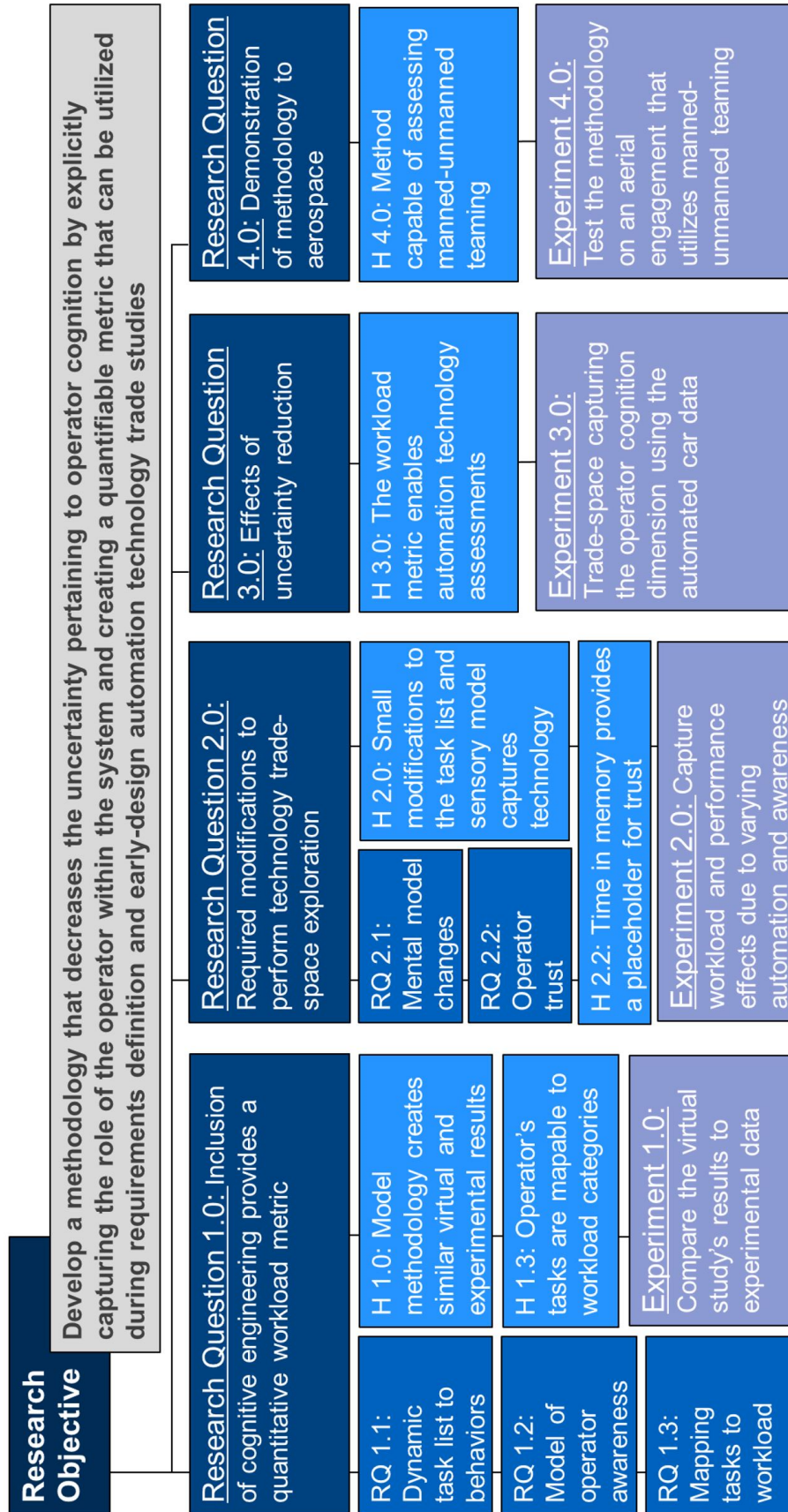


Figure 3.1: Summary of research questions, hypotheses, and resulting experiments.

found in literature is utilized for comparison. The work of Rahman et al [86] provides benchmark NASA TLX workload values for highway and city driving scenarios. This study was chosen because it focused on workload shifts due to overall situation complexity in real driving scenarios, which provides a relevant comparison to simulated environment shifts.

3.1.1 Experiment 1 Setup

This experiment captures the four identified critical pieces (an operations model, sensory model, dynamic task list, and workload model) to create a dynamic, traceable operator workload metric.

Operations Model

The operations model provides context for system analysis by capturing the environmental features encountered when driving in a city and on the highway. For driving workload, the road network and background traffic represent the operations model. The environments capture by the operations model constructs a representation of the driving study conducted by Rahman et al [86] (since the data will be used for the comparison). Paxion et al [81] discuss the different element categories that affect driving complexity: road design, road layout, and traffic flow. This experiment focuses on workload variations due to traffic flow as this category is variant throughout a model. Elements of road design and road layout are captured through the literature-guided, geographic specification of the road networks and characteristic traffic speeds based on the environment.

Starting with the specifics for the city operations model, Rahman et al [86] utilized the definitions from Paxion et al [81]. Rahman et al [86] defined their complex driving environment as having city roads (road design), speeds of 60-90 km/hr [37 to 56 mph], many curves and intersections (road layout), and a high density of traffic (flow). Using these characteristics as a guide, an agent-based, city environment is created in AFSIM.

The layout of the roads is specified utilizing geographic data from OpenStreetMaps [19] for downtown Atlanta, Georgia. This area is chosen due to its dense intersections and non-uniformity, matching the complex, city driving environment. The layout of the roads is shown in Figure 3.2, each lane represented by a pink line. Overall approximately 60 miles of downtown roads are captured. Intersections are shown by a gap between the lines. Overall the road network spans approximately five square miles and has 60 miles of roads. All roads in the network are assumed to be two lanes (one in each direction) with a posted speed limit of 30 mph. Although the city speed limit of 30 mph is slower than that utilized by Rahman et al [86], it was unclear the layout for the roads that they utilized therefore it was assumed that the utilized road layout should drive the speed limit typically experienced in downtown Atlanta. At each intersection along the main route there is a stoplight that stops traffic in either the north-south or east-west direction. The length of the green light is determined using a normal distribution with a mean of 60 sec and standard deviation of 45 sec, and a min green time of 20 sec and a max of 120 sec. The 60 sec timer was chosen based on the National Association of City Transportation Officials recommendation that urban lights be between 60-90 secs, while the timing distribution was added to contribute variations from adaptive signaling currently utilized in cities [16]. The placement of the stoplights is shown in Figure 3.2 represented by the yellow dots. The effect is best shown in the expanded view where a number of background cars are shown stopped to the north of the intersection. This background traffic is shown in blue and navigates the road network based on a simple rule set.

The background traffic's location and speed are randomly generated during initialization of the model. Each car is given a desired speed based on a normal distribution with a mean of 30 mph (the specified speed limit) and a standard deviation of 3 mph. The background traffic only responds to the "driver" car and stoplights. This simplification is necessary for runtime, because having all of the agents detect and respond to all features in the environment had an exponential impact on runtime. When a background car approaches

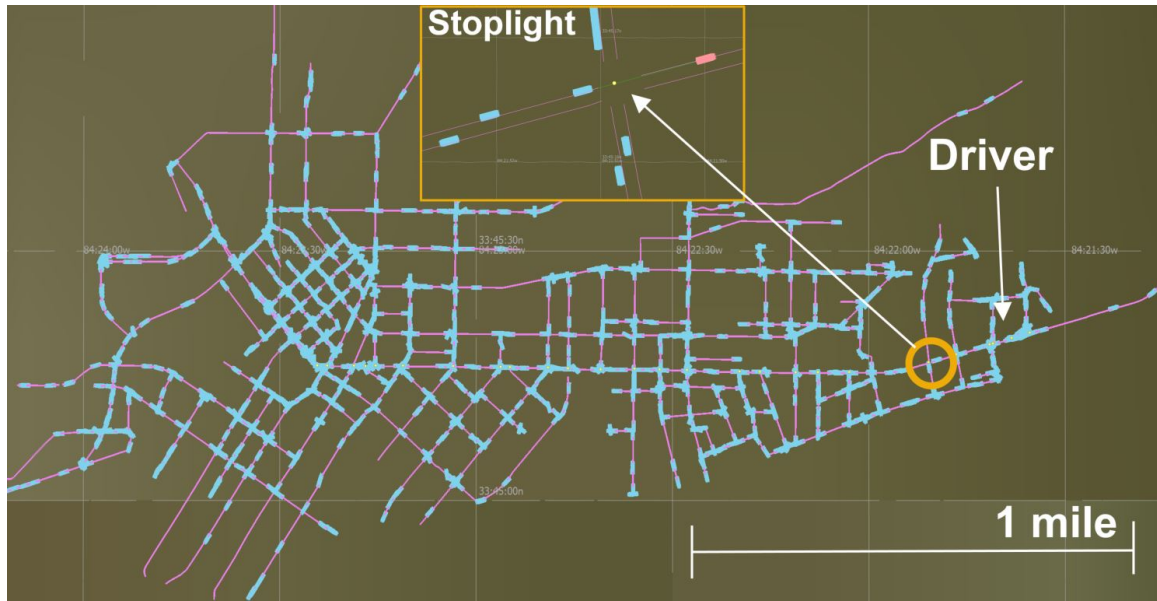


Figure 3.2: Layout of downtown Atlanta city roads in AFSIM

an intersection, it evaluates the stoplight. If the light is red for the car's current direction of travel then it stops and waits for the light to turn green. Additionally, if a background car approaches the driver's car from behind, it matches the driver's speed until the driver is no longer in front of it (ex: one of the two vehicles turn). The cars continually and randomly traverse the road network throughout the duration of the simulation making left-, right-, and u-turns. The operations model changes for each environment, so this defines the city environment however the highway environment must be separately created in AFSIM.

Rahman et al [86] defined their simple driving environment as having highway roads (road design), speeds of 110 km/hr [68 mph], an overall straightness (road layout), and low density of traffic (flow). Again these characteristics are utilized to specify a highway road network within AFSIM. Overall the road network spans approximately fifty miles and has 563 miles of roads. The highway that the driver starts on is a four lane highway (2 each direction), while the primary artery is eight lanes (4 each direction). All highways are assumed to be divided. The lanes run in parallel with an approximately 3 meter spacing between lanes. The multiple road architectures are specified to understand workload variations due to number of lanes and represent the variability due to highway design. The

posted speed limit of all roads is 70 mph, which is in line with that of Rahman et al [86]. Therefore, each car is given a desired speed based on a normal distribution with a mean of 70 mph (the specified speed limit) and a standard deviation of 5 mph. The background traffic has the same basic behaviors as those for the city environment (slowing behind the “driver”, random start point, and continual navigation). The highway background traffic changes lanes throughout the simulation.

The proof-of-concept for driver workload quantification compares the workload changes due to traffic flow and road type differences between environments. Both the highway and city environments contain simplifying assumptions for some of the real-world complexities involved in driving. The environments are assumed to be a homogeneous mix of vehicles (lacks semi-trucks, bicycles, pedestrians), does not include additional tasking from signage, and assumes no road layout changes such as merge lanes and speed changes. The environments are kept similar by controlling the amount of traffic represented. Both environments use a flow rate of approximately 1,000 cars per hour per lane. The number is constant for the city scenario, but the highway scenario has a variation between the two 8 lane highways. The first highway segment traversed has 90% of the flow rate and the second segment has 110% of the flow rate. This was done to demonstrate the difference in the workload and awareness models due to changes in flow density within the same simulation. All flow rates are chosen such that they remain under the free-flow rate of 1,300 cars per hour per lane for a 70 mph highway, while in between the typical saturation flow (600 to 1,800 veh/hr/ln) for signalized intersections [16]. With the background traffic in place, there is an environment to assess driving, however, it lacks the breakdown necessary for evaluating driver workload. None of the current agents monitor when and why an action is taken. Therefore, a more detailed agent, labeled “driver” in Figure 3.2, is required that is capable of performing detailed tasks based on its understanding of the environment and reporting these workload values.

Dynamic Task List

This virtual experiment seeks to quantify the intrinsic dependence between operator workload and his environment. Therefore it is important to take an in-depth look at how and why an operator performs actions within his environment while driving. Four primary driver actions are captured: navigation, road monitoring, changing lanes, and speed control. These tasks represent the beginnings of a workload model, enabling a basic comparison of workloads between environments and allowing for the inclusion of automation technologies in Experiment 2. Another considered driver action was maintaining lane, however, this task is similar for both environments and is a function of road geometry and speed more than traffic flow. The four driver actions are controlled through the agent's behavior tree.

Each action's branch on the behavior tree is created through the utilization of dynamic task lists. The dynamic task list connects the operator's goals, actions, and awareness elements. An example for changing lanes was developed by the National Highway Traffic Safety Administration and was discussed above in Figure 2.20) [62]. This task list and similar breakdowns for others actions are utilized to capture each action or behavior on the tree. The resulting behavior tree utilized by the "driver" is shown in Figure 3.3 and remains constant for both highway and city driving (a requirement if trying to compare workload due to variations in environment). The arrows help describe the flow within the behavior tree. A single arrow describes a sequence of nodes performed left to right until one fails (e.g. find roads to monitor cars), while stacked arrows show parallel nodes (e.g. change lanes and speed control). When in parallel, both nodes will be performed regardless of the first one's success or failure. Lastly, a selector node(e.g. speed) means each node will be evaluated and only one will be performed (if any). Starting at the top-left of the tree, the action of finding roads determines the "driver" agent's path along the road network. The agent's route planning is handled at the beginning of the scenario based on the start location and a desired ending location, while minimizing road distance. The agent determines at each intersection which road to follow based on the planned route. Between intersections

this action is not required. This action could be expanded to add processing time for finding alternative routes to avoid stoplights or minimize time if traffic is encountered. The next action is monitoring the car, with a leaf for determining the lead car. Monitoring the car assesses the “driver” current workload by quantifying the number of cars currently being monitored by the driver. From this list of cars being monitored by the driver, a lead car is determined (if applicable). The lead car must be in the driver’s current lane and perceived by the “driver”. If a lead car is determined, then it is utilized by the other behaviors.

The most detailed driver action is the behavior to change lanes. Lane merges can only occur if they are appropriate under current conditions. The “driver” does not accelerate or decelerate to create a suitable gap, the speed control actions are a separate behavior branch. This simplification is considered an adequate assumption because 91% of lane changes can be considered low severity and urgency [62]. Additionally, the effort for lane centering is not being tracked, so these tasks are also left off the behavior tree. Changing lanes starts with an evaluation of lane conditions. As shown in the dynamic task list (Figure 2.20), this starts with consideration, scanning traffic, and checking assumptions. The “driver” considers changing lanes if: 1) the lead car is impeding the “drivers” desired speed [changing lanes to go to desired speed], or 2) a lane is available on its right side [car moving to right lanes for general travel], or 3) nearing a merge point so needs to move to the right lane. If there is a desire to change lanes based on these criteria, then a traffic scan occurs. The traffic scan determines the suitability of neighboring lanes. If attempting to overtake a car, the “driver” scans the left and right lanes (checking both for applicability), otherwise for moving right or prepping to merge only the right lane is checked. If the lane is traveling faster than the “driver”’s current speed then it is a suitable candidate for the changing lanes, which results in a list of available lanes. This list is stored in memory and controls the awareness actions for the remainder of the lane merge. Based on the lane list created, it is time for the “driver” to evaluate the desired lane. In the behavior tree shown in Figure 3.3, this starts with the leaf node blind spot. Blind spot is an over-the-shoulder check of

the side of the vehicle, more detail on the specifics can be found under sensory model. If the blind spot is clear, then the tree proceeds to checking the mirror; however, if it not clear then the lane is determined unavailable and the other lane is checked (if it is in the memory list). A similar check for lane obstructions is conducted for the mirror. If the evaluation determines that the lane is clear then the final node is executed, which is the physical lane change. The requirement to maintain a safe distance between cars is handled separately by the speed control branch.

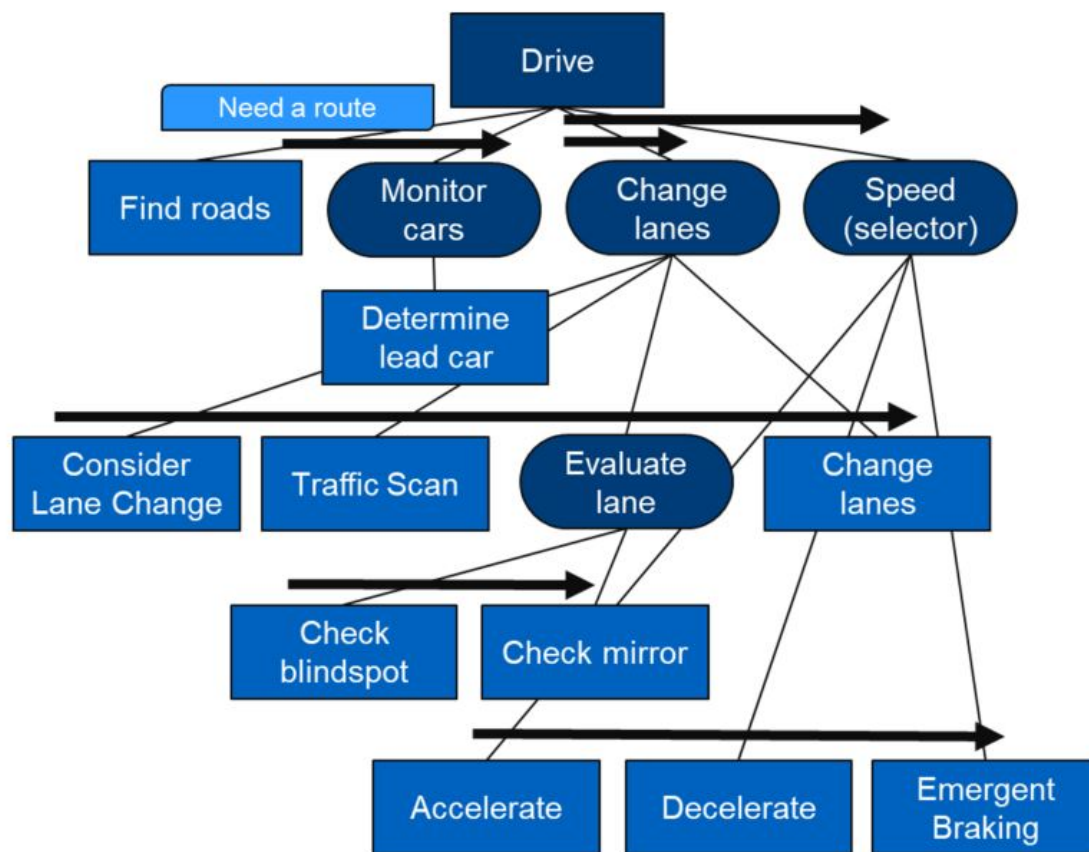


Figure 3.3: Representation of the behavior tree for driver agent showing the four primary branches: navigation, road monitoring, changing lanes, and speed control

The last primary behavior branch on the tree specifies how the operator controls the system's speed. As shown in the NHTSA dynamic task list and as experienced while driving, the primary purpose or goal of the speed controller is to ensure a safe following distance or resume a desired speed. If the car monitoring behavior determines a lead car, then the

goal of the speed control behavior is to maintain a gap between the “driver” and the lead car. The desired time gap is three-seconds based on NHTSA recommended safe driving practices [90]. If the “driver” gets closer than three seconds it will decelerate (the middle leaf on Figure 3.3), and if it gets further than four-and-a-half seconds then he will accelerate (the left leaf on Figure 3.3). The time between these provides an operational gap when following a lead car to ensure that the model is not always accelerating or decelerating. Acceleration and deceleration are controlled by changes in gap rather than using the lead car’s speed. This was done to more accurately represent how an operator controls speed without knowing the speed of the lead vehicle. The “driver” sets the car’s speed such that the gap remains constant based on current conditions. Therefore, if the lead car changes speeds this must be readjusted. If there is a sudden change in speed by the lead car or a car merges in front of the “driver” such that the separation distance is less than one-half of a second, then emergent braking is performed (the right leaf under speed control). Emergent braking works similar to deceleration, however, it sets the braking to twice the gap’s closing rate. Lastly, when there is no lead car, the acceleration node may be utilized if the “driver” is not going his desired speed, then he returns to his desired speed.

The four primary branches on the behavior tree provide an insight into the detailed interactions between an operator and his environment. It clarifies the why and how for actions occurring. The behavior tree is executed once every 0.5 sec, which was assumed to be frequent enough to capture variations. This update interval determines when behaviors can change, however, the system does perform specified actions between intervals such as accelerate and decelerate to the determined levels. This “driver” captures common driving action for which automation technologies have been built to replace, such as adaptive cruise control for the speed branch and autopilot for the changing lanes and speed branches. These basic actions represent a building block for more complex models. This behavior tree allows insight into the “drivers” actions, however, for it to properly interact with a changing environment, it must have a sensory model that can capture the operator’s situational

awareness throughout the operational scenario.

Sensory Model

The sensory model is a critical piece to capturing the operator's awareness, rather than the system's awareness. The sensory model takes into account where the operator is focused to decide which information is acquired and stored in the "drivers" memory. This sensory model focuses on level 1 (perception of elements) and level 2 (comprehension of current situations) situation awareness, as defined by Endsley [29]. Level 1 is captured through the location and heading of cars perceived by the driver, while level 2 data is converting that information into an understanding of each car's relative speed and closing distance. Level 3 SA (projection of future status) is difficult to capture without modeling more cues utilized by the driver (brake lights, turn signals, head direction of the other driver, etc.).

The first step to defining the sensory model is understanding which pieces of information does the operator seek and acquire. For this operational environment, the pieces of information are: left side, right side, left mirror, right mirror, near front, and far front. The side information represents an operator looking over his shoulder, while the mirrors represent a glance to understand traffic behind the car. Figure 3.4 shows all six of the sensor channels represented on the vehicle, which have been assumed to have the following characteristics. The side channels have a range of 16 feet and a field of view from 20 degrees to 160 degrees. This represents being able to see along the side of the car for one full lane in range (lane widths are 12 feet for highways in the United States [77]). The mirrors assess behind the car in a neighboring lane and are capable of seeing 53 feet. The exact distance checked by a driver was difficult to find, so the 53 feet represents a three second gap (based on a 10 mph speed difference). This distance is assumed to be an acceptable mirror range when deciding to change lanes because it is in line with the NHTSA's three-second safe following distance metric [90]. Lastly, there are two forward looking sensors (near and far). The near channel represents scans from left to right to understand traffic

patterns in neighboring lanes for highway driving and cross traffic at intersections. It also fills in the gaps in the far forward sensor to determine travel speeds of neighboring lanes for determining desirability of a lane change. The near forward sensor has the same range as the mirrors (53 feet) and a field of view from -40 deg to 40 deg. The far forward range sensor has a field of view from -5 degrees to 5 degrees and is utilized to understand what is occurring further out in the “drivers” current lane of travel and speeds of neighboring lanes. It monitors everything within five seconds of the front of the car based on desired speed, so it changes range for city and highway driving. When in the highway environment, all channels ignore cars with heading differences greater than 45 degrees from the “driver”, representing a divided highway.

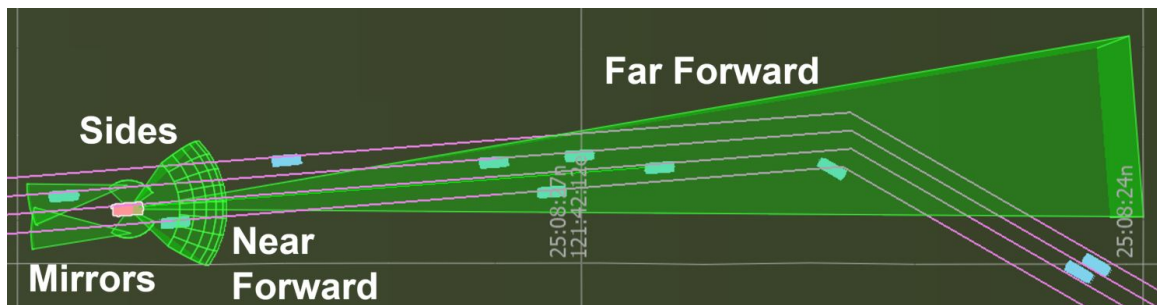


Figure 3.4: The six information channels for the sensory model: left side, right side, left mirror, right mirror, near front, and far front. The behavior tree helps determine which channels are active.

With the pieces of information available to the operator defined, the next step in the sensory model is defining when each channel is active and for how long. Channel activation is determined by the behavior tree. The operator focuses general attention to the forward channels, however, when required he will use the side and mirror channels to change lanes. Figure 3.5 shows the activation of the two forward channels utilized for general driving. This represents the current information being acquired by the driver and the information that is used to determine a lead car, desire to change lanes, and necessity to change speeds. The far forward channel remains active for the entire operational scenario. Since mirror and side detections are considered glances, it was assumed that the driver would maintain

some awareness of their current lane of travel, but would not have forward sweeps left and right while checking other sensor channels. Therefore, when the leaf node for checking blindspots is activated, the sensor channel for the appropriate side is turned on and near forward is turned off. Figure 3.6 shows this occurring for a desired left lane change. The “driver” is being slowed by the car in front of him, so he considers a lane change to improve his speed. The “driver” scanned the surrounding traffic pattern based on the forward channels and determined that the left lane is moving faster. After the traffic scan determined the left lane is suitable, the next step in the behavior tree is a lane evaluation. This lane evaluation takes time to gain the necessary situational awareness. A summation of experiment findings in NHTSA’s “A Comprehensive Examination of Naturalistic Lane-Changes” discusses the different studies looking at mirror glance times and distributions [62]. From this summation, the following lognormal distribution parameters are chosen: the glances to the left mirror are 1.1 seconds with a standard deviation of 0.33 seconds, and glances to the right mirror take slightly longer at 1.21 seconds with a standard deviation of 0.36 seconds. These same parameters are assumed for left and right over the shoulder checks since data could not be found. Therefore, when the “driver”, activates the evaluate lane component of the behavior tree, these times are utilized to determine the time to complete each action. If attempting to merge left as shown in Figure 3.6, when the blindspot leaf node is activated a random number is drawn from the lognormal distribution to determine how long it will take to conduct the side check. The behavior tree may only proceed to check mirror once this time has elapsed. Check mirror then does another random draw from its lognormal distribution to determine the time required for awareness. During this time and only for the specified duration, the “driver” is capable of adding the cars to memory for this sensor channel.

“Driver” memory is continually refreshed with current observations. Cars in sight by the active channel are continually refreshed with heading and location data. Any car not currently being observed is assumed to be remembered for five seconds. If it is not seen

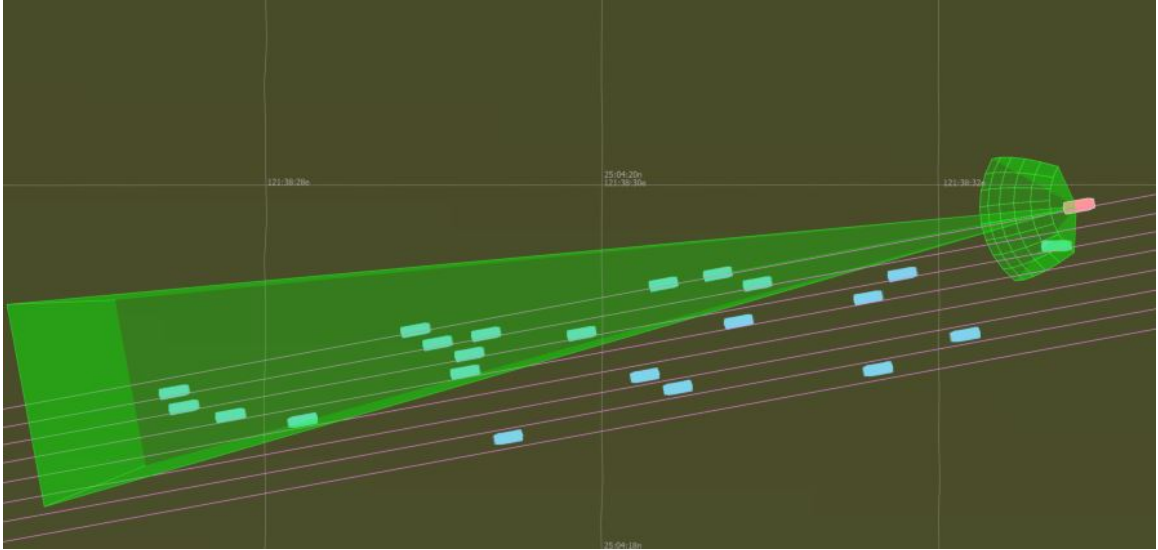


Figure 3.5: The two active information channels for general driving (near forward and far forward). The near front represents eye scans left to right, while far front the current lane of traffic further out for speed control.

again, then the car is dropped from memory and is not used in any further calculations until it is rediscovered. This representation is in line with Endsley’s research, where an operator seeks out the information required as needed to fill in knowledge gaps [32]. Adding the sensory model to the dynamic task list creates a “driver” that is now capable of performing detailed tasks based on decomposed operator actions which are guided by his perception of his current environment. However, for this to provide insight into the operator’s workload, the operator’s actions and awareness must be mapped to the NASA TLX categories.

Workload Model

The workload model is the last critical piece of the proposed methodology and provides the metrics by which automation technology may be assessed. This methodology measures three workload metrics: mental demand, temporal demand, and performance. Mental demand represents the workload due to the operator’s perception and tracking activities [47]. Based on the current operational scenario, it was assumed that the number of cars in the “driver’s” memory provides a sufficient corollary to mental demand. These cars represent



Figure 3.6: Image depicting the lane evaluation behavior. The driver observes a slow car 200ft ahead in its current lane and has a desire to merge into the left lane. The left side channel is active, shown in green, while the near forward channel is inactive. The far forward channel is also active during this behavior, but is not shown for clarity.

the number of items to which the operator is actively paying attention at each discrete time step within the simulation, and it is a consistent measure of the current task load regardless of the environment.

To further aid in the discussion of these metrics, Figure 3.7 shows the measured workload for a 600 second simulation of highway driving. The yellow line in Figure 3.7 shows the instantaneous changes for cars in memory over time in the simulation. Capturing the variations in mental demand enables time-based workload analysis, where isolated segments of the scenario can be assessed. This line shows a peak mental workload at 17 cars in memory for two brief instances, but both occurring on the highway with 4-lanes in each direction and a high density. Low mental workload levels are shown for the 2-lane highway. The other lines (speed, lane changes, and intersections) are shown to provide context

for what is occurring in the operation scenario at those time steps. Figure 3.8 converts the instantaneous mental demand into time at each workload-level to capture the distribution of expected workload throughout the highway scenario. The mode for the mental workload is 6 cars in memory and occurred for 109 seconds. For the overall mental workload of a scenario, a time based average is taken using the following equation:

$$AvgMentalDemand = \frac{\sum_{i=0}^{MaxCars} time_{atWorkload} \times NumberOfCars}{time_{total}} \quad (3.1)$$

For Figure 3.8, this results in a mean mental workload of 6.55 cars in memory. This average mental workload will provide a comparable metric between numerous operational scenarios and existing human-based studies.

Temporal demand is the next NASA-TLX workload being captured. As a reminder temporal load represents the time pressure felt by the operator or the pace that the tasks are occurring [47]. Since it is a quantifier of pace, the mental workload's rate of change is utilized. Every time the mental workload changes (e.g. two cars in memory to three cars in memory) whether an increase or decrease, the time is recorded. The time between each change represents the pace of the workload. Therefore, the inverse of this time represents a temporal metric with higher numbers implying increases in workload (i.e. quicker time between changes to cars in memory). This is shown by the variations in the yellow line (i.e. cars in memory) on Figure 3.7. The temporal demand is directly correlated to these variations, with more frequent changes being tied to a higher temporal workload. The temporal variations throughout the simulation are shown graphically in Figure 3.9. The blue line shows the time at a fixed mental workload level (e.g. at a simulation time of 7 seconds the horizontal line at 27 correlates to a mental workload of 3 cars in memory for 27 seconds). During this time duration, the temporal workload is low at 0.04 [1/sec] (1/27 seconds). This figure also shows the rapid fluctuations in temporal demand based on traffic patterns.

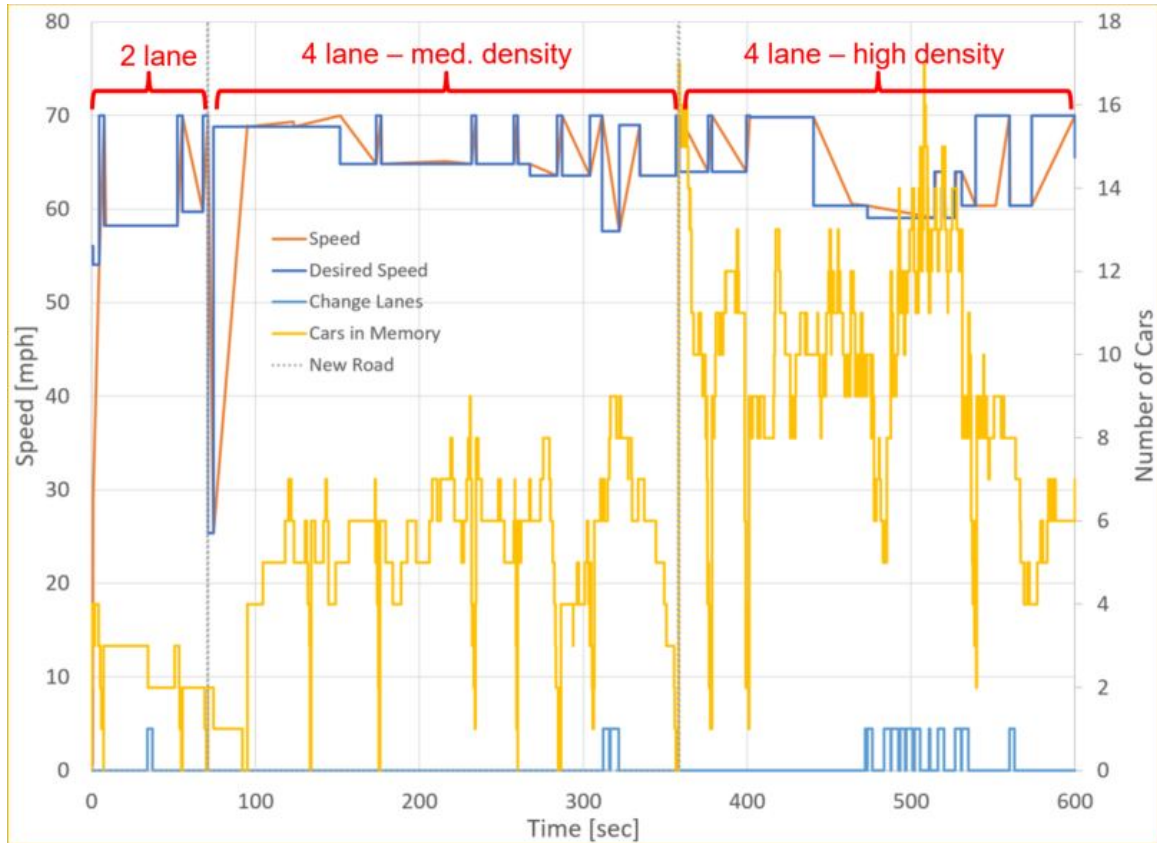


Figure 3.7: Workload metrics for a highway scenario with 8,000 vehicles. Flow rate is 1,000 cars/ln/hr for the two lane highway, 900 cars/ln/hr for medium density and 1100 cars/ln/hr for high density. Distance traveled was 10.7mi (desired 11.67mi) – 92%

A log scaling is used for the temporal demand axis due to the behavior of inverse functions. When multiple cars are added to memory at the same time or near simultaneously the load approaches infinity, which is a short coming of the current correlation between time and temporal load. For the simulation shown, a mental workload change occurred on average every 1.16 seconds, which leads to an average temporal load of 0.86 [1/sec]. Similar to above, regions of the operational scenario can be further analyzed to determine average temporal load for high flow conditions or other, specific simulated areas of interest.

The last workload metric of interest is the operator's performance. This measure provides feedback on the operator's and system's ability to attain the operational goals [47]. For driving, the difference between actual distance traveled and ideal distance traveled is used to measure as the performance metric. Ideal distance is the distance traveled if the

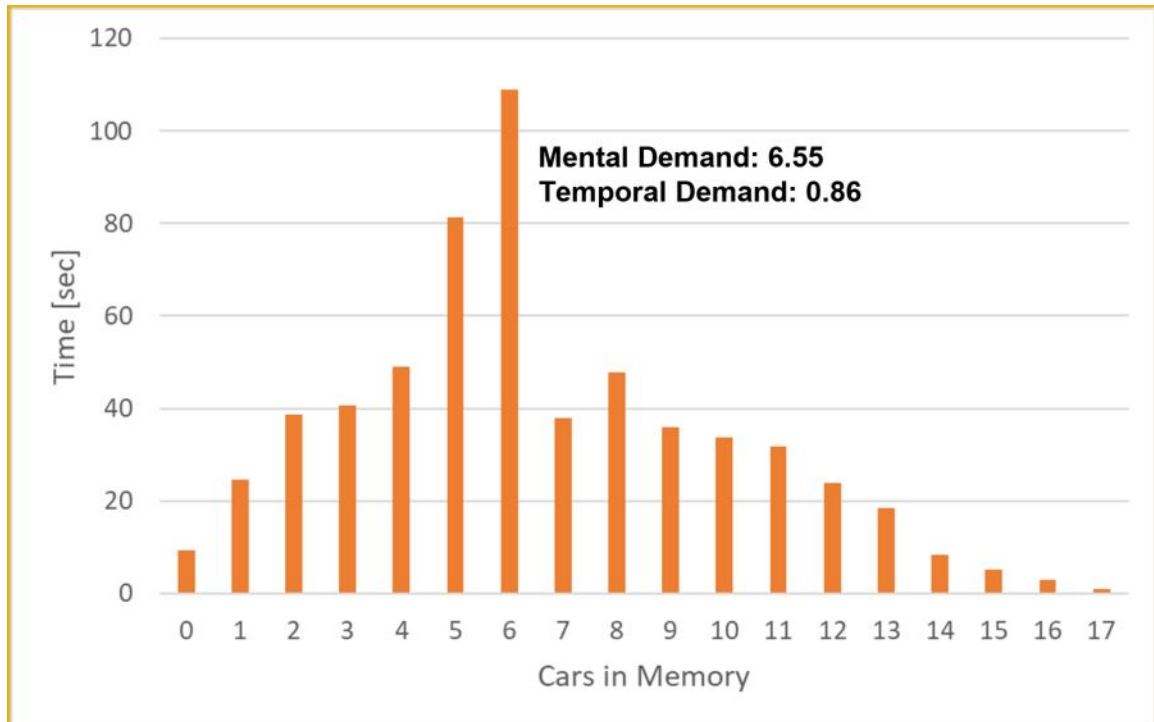


Figure 3.8: The time spent at each mental workload level for the highway scenario in Figure 3.7 (represented by the number of cars in memory). The time based average for mental demand is 6.55 cars in memory, and the temporal demand is 0.86 [1/sec] (based on the inverse of the average time for changes in mental workload)

operator was able to travel his preferred speed for the duration of the experiment (70 mph for highway driving and 30 mph for city driving). Distance traveled can be determined in the simulation by utilizing the time and speed data from Figure 3.7. This figure shows the duration that the “driver” drove at each speed, which is recorded every time a speed control behavior was executed in the dynamic task list. Desired speed represents the speed determined by the speed control behavior, and varies based on a lead car’s speed. From the current data, the “driver” traveled 10.7 miles in the 10 minute operations model. If the “driver” had been traveling at the preferred 70 mph for the entire time, he would have been able to travel 11.67 miles which correlates to a performance of 92%.

These three workload metrics provide the necessary components to compare to the results of Rahman et al [86] and assess alignment to their usage of NASA TLX. However, inclusion of a detailed task list and sensory model in an operations model is capable of

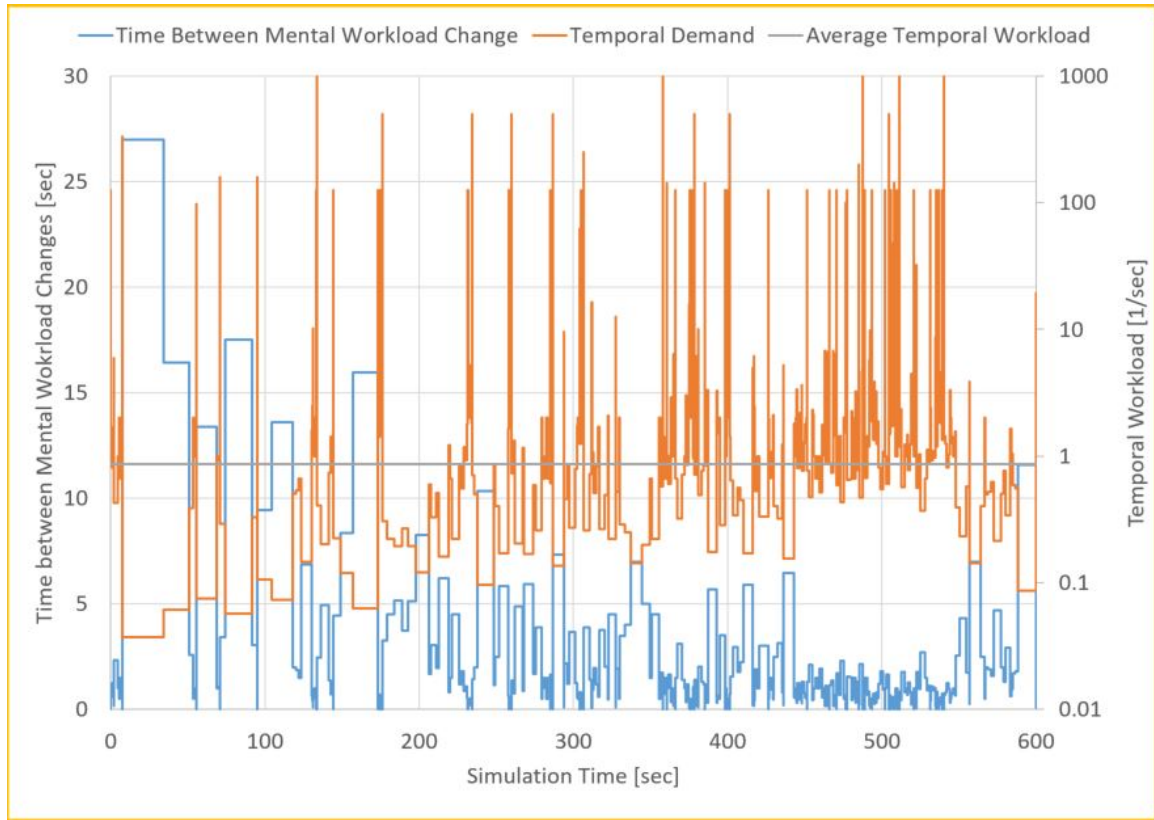


Figure 3.9: The times to change mental workload (the pace of task changes) and the related temporal demands for the highway scenario. The average temporal demand is 0.86 [1/sec] (based on the inverse of the average time for changes in mental workload)

providing further insights. Proper automation integration requires an understanding of the operator's actions and the frequencies by which they are conducted, or automation can create performance degradation [27] [12]. The addition of these model components, and the linkage to real-world cognitive studies, allows insights into the timing of operator actions and the workload at those instances.

3.1.2 Experiment 1 Results

With the four parts of the model defined and integrated, it is time to capture workload data and compare the results with existing studies. Figure 3.10 shows the “driver’s” actions and workload throughout the simulation. The gold line captures the instantaneous mental demand while variations in cares in memory show the temporal fluctuations. The operator’s

actions are shown as spikes throughout the simulation. Each spike represents the behavior being executed, while the width shows the duration of the activity. Lane change activities are the longest activity because it encompasses the whole duration that the operator is focused on that activity. The timer starts when the “driver” conducts the traffic scan and concludes when all lanes have been checked but no lane change occurred or when the lane change occurs. Looking further into the changing lanes action, the behavior tree dictates side checks then mirror checks. However, sometimes the side check fails (a car is there), so no mirror check is conducted. The left lane is more likely to be checked because of the presence of slow cars in the right lane after the traffic scan. Overall, this activity breakdown shows an operator who is involved in the driving task and continually using the cars in memory to perform these actions. This breakdown of the operators actions provides insight into changes created by the addition of automation technology. The action breakdown also provides a secondary point of comparison for the virtual results versus real-world studies.

The purpose of Experiment 1 is to test Hypothesis 1.0 which states, “the inclusion of a dynamic task list, sensory model, and workload model in an agent-based operations model (AFSIM) will enable a dynamic, traceable, and quantifiable operator workload measurement similar to those currently manually conducted”. Through this analysis Experiment 1 also tests Hypothesis 1.3 which states, “the detailed task list will map to different elements of the workload measurement techniques providing a dynamic, traceable assessment throughout the simulation”. The mapping between workload elements is determined acceptable if similar comparatives are found between the virtual and workload experimental data. Experiment 1 contrasts this virtual workload data for the city and highway scenarios with a 1,000 car flow rate then compares these differences to those found in literature. Variations in the highway traffic flow rate were also included in the scenario to enable sensitivities of the measured workload to be determined. A similar stochastic comparison is conducted using the original 1,000 car flow rate to examine the effects of car placement variations on workload. By running the virtual simulation multiple times, the random

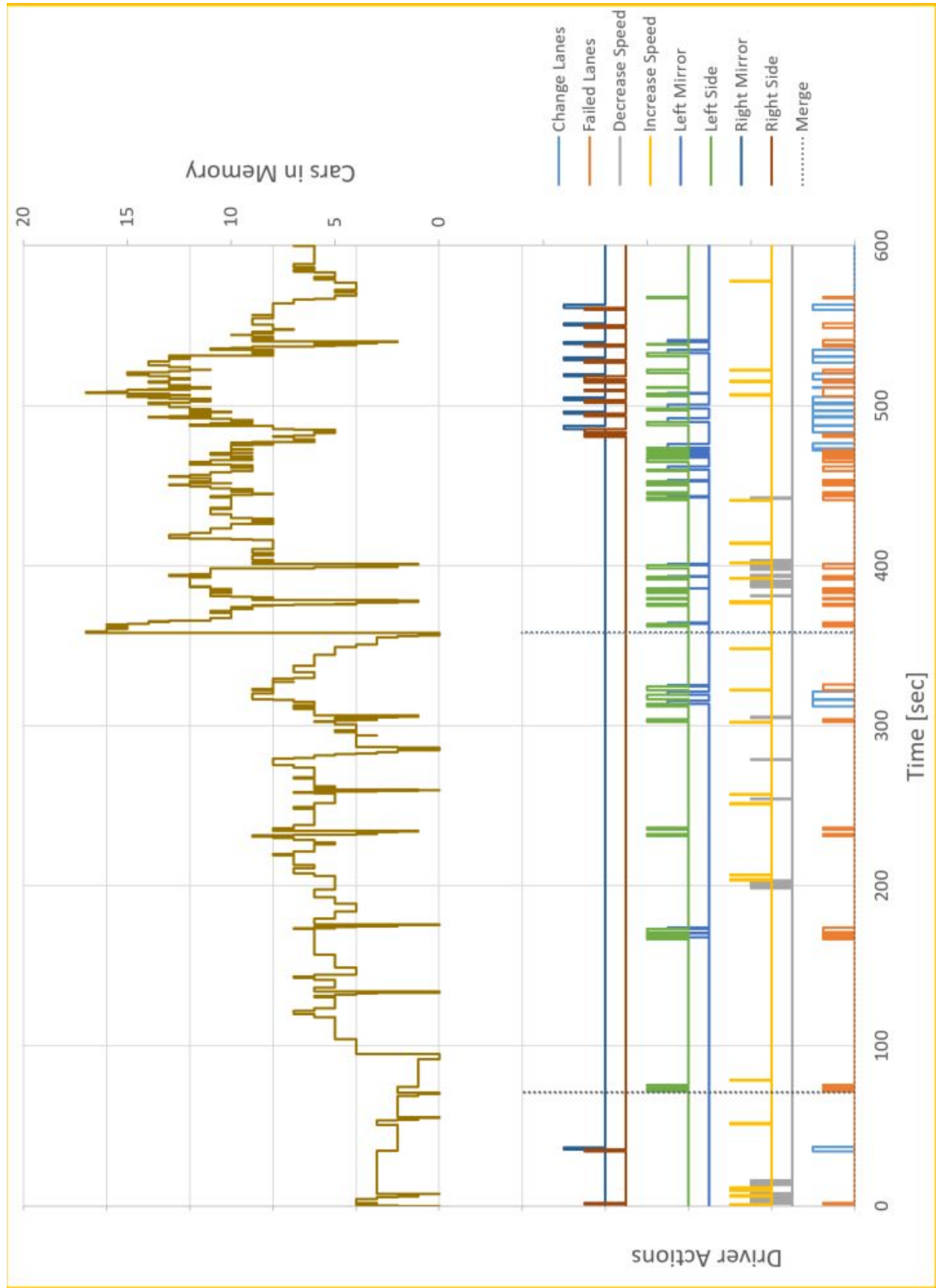


Figure 3.10: The actions taken by the operator throughout the operational scenario. The same 8,000 car scenario has been used. The width of the spikes demonstrates the duration of the action.

placement of the cars changes resulting in variations to the background traffic behaviors and effecting the “driver’s” awareness and actions.

Virtual City and Highway Data

The city and highway virtual results demonstrate the capability of this methodology to create dynamic, traceable, and quantifiable results. Experimental studies utilize a numerical scale for each NASA-TLX category at the end of the study. Therefore, each operator provides one number for the entirety of the operation. The virtual metrics allow further insight into these workload categories. The highway results are shown above in Figure 3.7 and Figure 3.8. In Figure 3.7 instantaneous mental demand is shown by the yellow line. Taking a time-weighted average of this demand creates an overall mental demand similar to that in experimental studies. This time-based binning of mental demand is shown in Figure 3.8, and an average mental demand of 6.55 cars in memory is computed. However, this continuous workload capture allows further traceable and quantifiable insights into the primary workload drivers and the variations in workload. Figure 3.7 shows that for the duration that the “driver” was on the two-lane highway the workload had a maximum of 4 cars in memory, while for the high density, four-lane highway it spiked up to 17 cars in memory for brief intervals. Overlaying driver actions, it can be seen that the majority of the “driver’s” lane changes are required when mental demand is high. Intuitively this makes sense because the “driver” is trying to resume his preferred speed when getting stuck behind slow cars. Although he is going slow on the two-lane highway, the behavior “traffic scan” fails to find a better available lane. This breakdown of mental workload allows actions to be compared with instantaneous workload and understood in context. This context is important as automation is added to answer teaming questions such as: ‘were the cars in memory not utilized for a long duration?’, ‘was a behavior performed after a long lull in memory usage?’, and ‘what mental workload was the operator under while performing this action?’. This continuous mental workload translates directly to traceable and quantifiable

temporal demand. The pace of the workload is determined by the frequency of mental demand shifts. This can be seen in Figure 3.9. The temporal demand is shown to be higher and have increased fluctuation with more cars in memory, which makes sense as there is a higher likelihood of new cars entering or leaving the “driver’s” awareness. The last comparative metric from the virtual study is performance. An overall performance metric is calculated as 92% utilizing the distance traveled of 10.7 miles and an ideal distance traveled of 11.67 miles. However, utilizing Figure 3.7, the performance at each time and location in the scenario can be better understood. The driver continually attempts to maintain his preferred speed of 70 mph. However, the graph shows the operator’s desired speed based on the current environment (e.g. a lead car traveling slower) and the vehicle’s current speed. Every time that the desired speed is below 70 mph the driver is performing under expectations. Figure 3.7 shows the highest performance for the four-lane medium density environment, while the two-lane and four-lane high density have lower performance. This is because for the four-lane configuration the “driver” has more options for changing lanes when desired, however at high-densities it is difficult to find a lane at the desired speed. The 66 mph desired speed is an allowable speed based on the tolerance specified in the change lanes behavior. This tolerance could be changed such that the “driver” always seeks out a 70 mph lane. Breaking down the new virtual results shows the applicability of the new methodology to create metrics that are quantitative and traceable. The desire for a dynamic capability can also be seen in the highway example due to different configurations, however, it can better be seen when contrasting the highway results to the city results.

The city environment maintains the same components for the dynamic task list, sensory model, and workload model, but changes the road structure and preferred speed. The city environment assumes only one-lane (each direction) roads, so lane changes are not an option; however, it does include stoplights. The results for the city environment are shown in Figure 3.11 and Figure 3.12. Similar to the highway model, in Figure 3.7 instantaneous mental demand is shown by the yellow line. Taking a time-weighted average of this demand

yields an overall mental demand. This time-based binning of mental demand is shown in Figure 3.12, and an average mental demand of 6.19 cars in memory is computed. The city scenario had a similar range for mental demand from 0 cars in memory to briefly containing 19 cars in memory, however, it does not show the central tendency behavior seen with the highway environment in Figure 3.8. As the “driver” approaches the center of the city environment there is an increase in mental demand as congestion is encountered around the stoplights. This congestion dissipates near the edges of the city as the car clears congestion. Overlaying driver actions, the variation in mental workload at each intersection is noticeable. This mental workload breakdown creates a quantifiable and traceable metric for mental workload throughout a busy environment, compared with the experimental one average number, a necessity for understanding when and how automation should interact with the operator’s actions. From the methodology, the time variations in mental demand create the temporal demand (as shown in Figure 3.13). The temporal demand for the city scenario is higher than the highway scenario, even though the average mental demands are similar. This shows that there are more cars continually entering and leaving the “driver’s” memory in the city scenario creating a constant temporal load as awareness must continually be attained. A comparative performance metric is calculated as 53% utilizing the distance traveled of 2.63 miles and an ideal distance of 5.00 miles. The ideal distance is calculated assuming the drive is able to maintain 30 mph for the entire 10 minute simulation time. Figure 3.11 shows the instantaneous performance at each time and location in the scenario. The driver is attempting to maintain 30 mph for the city scenario, however, often ends up below this speed due to stoplights and a lead car. The congestion and control elements in the city environment continually hinder the “driver’s” desired performance, whereas in the highway environment the “driver” can attempt avoiding the performance degradation from slow lead cars. The capability to continuously capture mental demand, temporal demand, and performance in multiple environments shows a dynamic, traceable, and quantifiable operator workload metric not traditionally captured in operational analysis. However, this

methodology is really only useful if the methodology creates a metric that is comparable to experimental results, therefore the next step is comparing the virtual data to existing experimental data.

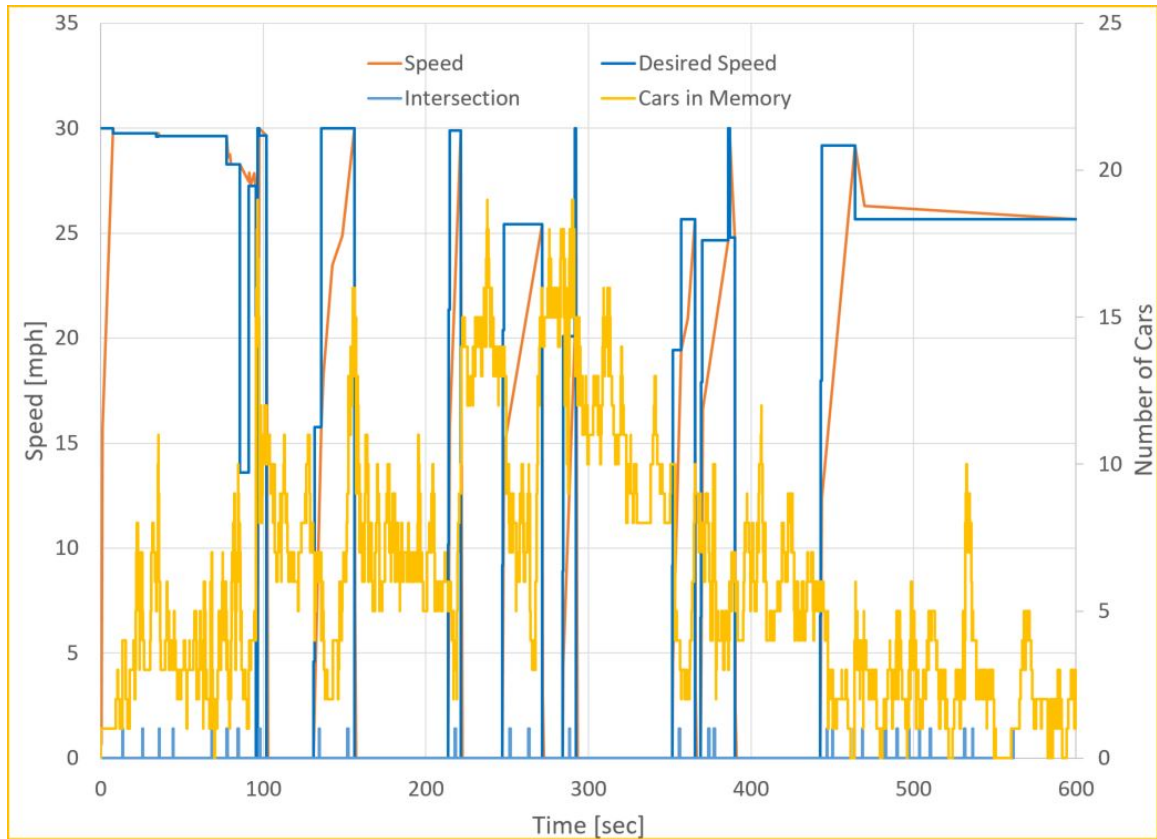


Figure 3.11: Workload metrics for a city scenario with 2,000 vehicles. Flow rate is 1,000 cars/ln/hr throughout the road network. Distance traveled was 2.63mi (desired 5.00mi) – 53%

Virtual versus Experimental

The first point of comparison for the new workload metric is assessing the measured differences between environments for the virtual model versus those done experimentally. For this comparison, the findings from Rahman et al [86] are utilized since they looked at the effect of environment shifts on operator workload. The results from their study are shown in Figure 3.14a. Their study found similar mental demand between city and highway environments, considerably higher temporal demand for the city setting compared with the

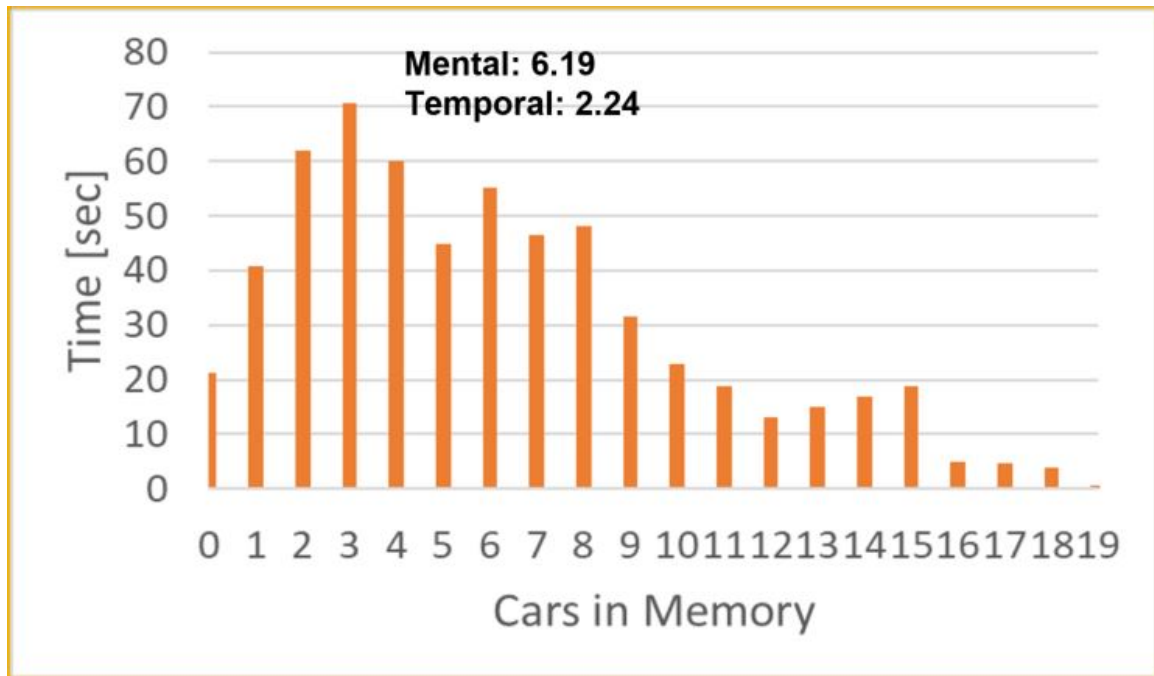


Figure 3.12: The time spent at each mental workload level for the city scenario in Figure 3.11 (represented by the number of cars in memory). The time based average for mental demand is 6.19 cars in memory, and the temporal demand is 2.24 [1/sec] (based on the inverse of the average time for changes in mental workload)

highway setting, and a noticeably higher performance for highway driving. Experimental results use the NASA-TLX score as an overall workload metric, not measuring instantaneous variations, therefore these results will be compared to the average values found in the virtual study and discussed above. Also, since the NASA-TLX scales are qualitative, it is difficult to compare results directly. Therefore, normalized differences and general trends between the two data sets are utilized. The bottom of Figure 3.14b shows the normalized comparisons between the virtual and experimental results. The normalization is done for each workload category by dividing the highway averaged workload by the city averaged workload. City workloads are always 1, while the highway data varies after being divided by its respective (virtual versus experimental) city value. The comparisons after this normalization shows promising results for the new methodology. Mental demand for both the virtual and experimental data shows similar workload levels for both city and highway environments. For the study data, the highway workload was 88% of the city workload,

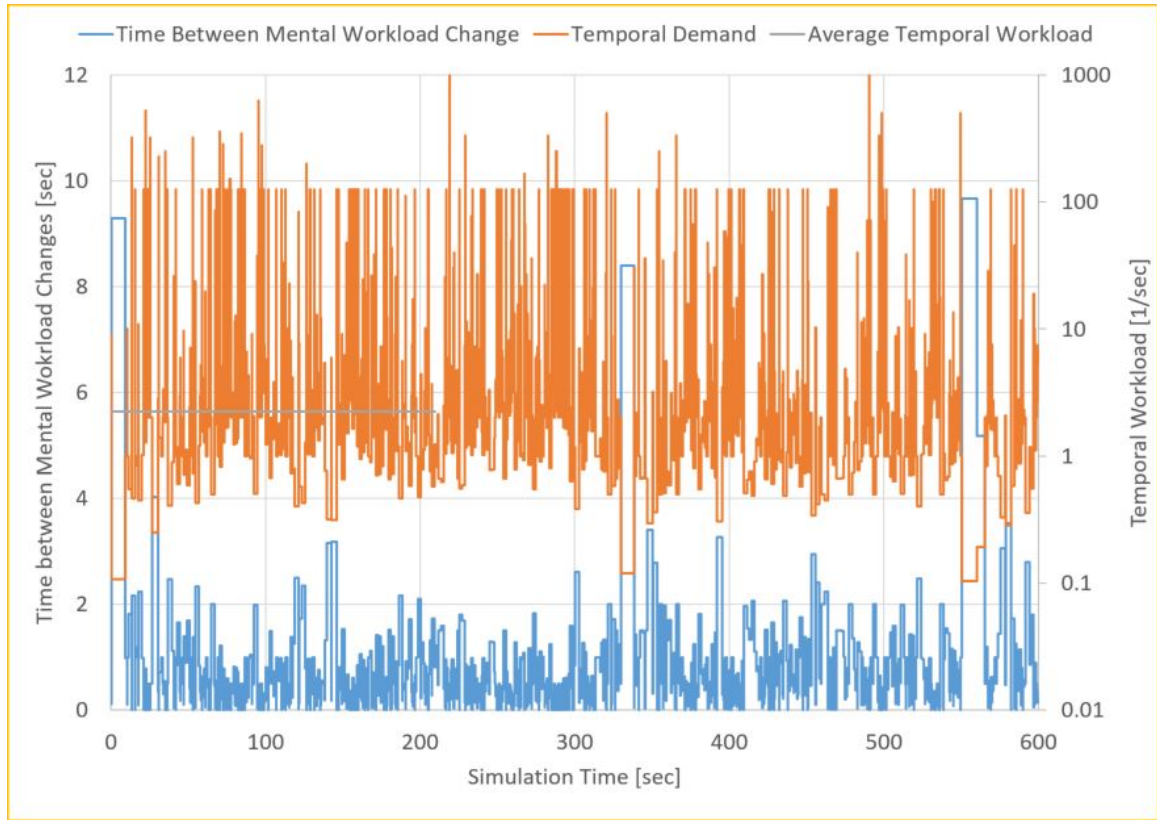
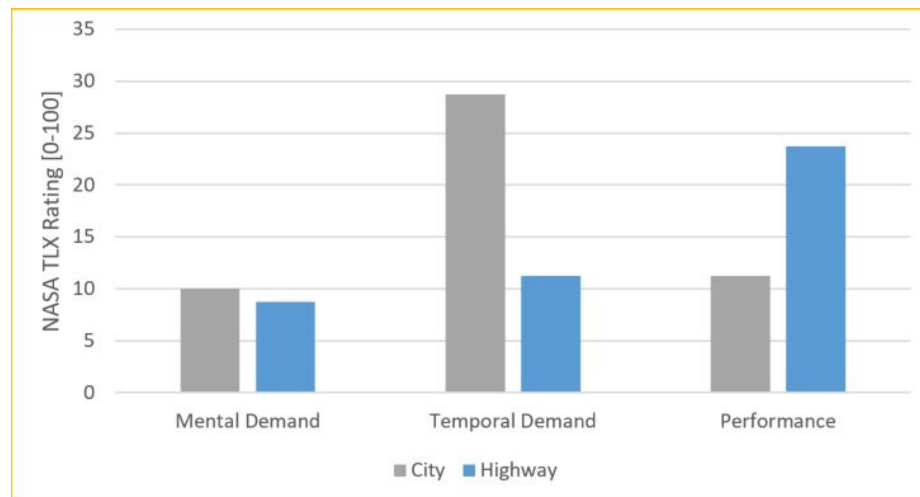


Figure 3.13: The times to change mental workload (the pace of task changes) and the related temporal demands for the city scenario. The average temporal demand is 2.24 [1/sec] (based on the inverse of the average time for changes in mental workload)

while for the agent-based model (ABM) is 106%. This may be explained by the lack of additional visual tasks such as from sidewalks, which are typically higher in city environments. The ABM showed almost exactly the same difference between temporal demands. The highway temporal demand is 38% of the city temporal demand for the virtual study, whereas it was 39% for the experimental study. Lastly, the performance for the virtual highway environment is 168% of the city performance, whereas the experimental results shows a 211% improvement. Overall, although there are differences, all three categories show similar trends between virtual and experimental data. Where a small differentiation was observed in the experimental data, a similar difference is observed virtually as shown by mental demand. The same trend is observed for large differences, captured by variations in temporal demand and performance between different environments. The most starkly

different category between virtual and experimental results is performance, however, this may be partially attributable to the higher city speeds utilized by Rahman et al [86]. Higher preferred speeds would correlate to a reduced performance due to stoplights and congestion. Since driving is such a variant activity, it is difficult to exactly align these metrics; however, the similar trends between virtual and experimental results provides credibility to this methodology.



(a) Experimental data from Rahman et al. [86] assessing driver workload under different environmental conditions: highway [simple], and city [complex]



(b) Comparison of workloads from this user-based study to the agent-based model. Each workload category contains the highway data normalized by each study's respective city data.

Figure 3.14: Comparison between experimental and virtual NASA TLX Workload Variations.

One of the benefits of the bottom-up modeling approach utilized for agent-based modeling (ABM) is its ability to capture emergent phenomena [9]. In many cases this capability is necessary because the data required for top-down modeling is unavailable. However, in the case of driving, there are some top-level metrics which have been experimentally quantified to assess the bottom-up approach being utilized in the definition of the dynamic task list and sensory model. If there is alignment between the emergent behaviors and the experimental measurements, then this provides more validation for the operator-focused methodology. For this comparison, the highway data set is utilized since it includes the more detailed action of changing lanes, and this action is well studied in literature. The two top-level metrics being compared are the number of lane change actions and the amount of time spent not looking forward. First, the number of lane changes in the virtual study is compared with experimental data. From an NHTSA study on naturalistic lane changes, a sedan driver performs an average of one lane change every 2.4 miles [62]. Using this lane change rate, the virtual highway study would be expected to have 4.5 lane changes over the 10.7 miles traveled. However, the simulation resulted in 13 lane changes, which is three times higher than the experimental data. Figure 3.7 shows the virtual data is primarily broken into four clusters, with the high repetition of lane changes occurring between 473 seconds and 535 seconds. As shown in the speed data, during this interval the “driver” is continually trying to find a lane capable of traveling its desired speed. As neighboring cars change speeds slightly, it moves back and forth between lanes accounting for 9 of the 13 lane changes, an action unlikely to be taken in traffic. Therefore, although there are more lane changes in the virtual study, this is likely attributable to the structure of the dynamic task list and could be overcome by disallowing repetitive lane changes (going from lane 1 to lane 2 then back to lane 1 in a short period of time). This action was not taken to not overly constrain the emergent behavior since actions are only conducted based on neighboring lane. An improved sensory model capable of ‘seeing’ and deciding 2 lanes over or “driver” memory for why the last lane change occurred and from which lane could likely

overcome this issue.

The next point of comparison is the time that the “driver” spends focused on the near front sensor, instead of the sides and mirrors. From the action list in Figure 3.7, the time spent looking at mirrors and side channels was calculated as 138.2 seconds. This accounts for 23% of the simulation being spent on non-forward tasks, resulting in 77% attention towards the front channel. This 77% is higher than the 67% attention towards the forward roadway found experimentally [9]. The fact that the current virtual model overestimates attention to the roadway is expected. It can be attributed to the basic dynamic task list and resulting behavior tree. The operator’s attention towards road signs and secondary tasks (changing radio stations) is completely ignored from the current model, by default this time is added to the general forward channel. While there are differences between the experimental data and virtual simulation results, these differences can be easily attributed to simplifications in the model as compared to the complexities in real-world driving. Considering the assumptions made in the virtual simulation, comparing the virtual methodology average workload scores and top-level metrics to existing experimental data proved promising.

Based on the results obtained in the benchmarking study, both Hypothesis 1.0 and 1.3 are accepted. Hypothesis 1.3 claims that there exists a mapping between operator tasking in a virtual environment and a quantitative workload metric. Experiment 1 captures the operator’s actions and awareness elements then maps these to the NASA-TLX scales. Hypothesis 1.0 claims that the methodology of utilizing a dynamic task list, sensory model, and workload model in an operations model creates a dynamic, traceable metric. Experiment 1 combines these elements to create a software agent that focuses on the operator and guides agent actions based on operator awareness. This operator-focused agent then captured the workload variations between a city and highway environment. Comparing workloads from these two virtual environments to data found in literature found good alignment between workload differences leading to the acceptance of the Hypothesis 1.0 and the methodology

elements regarding operator-focused agent creation. However, prior to adding automation technology to the existing model, it is important to understand the sensitivity of the model to specified flow conditions and its stochastic behavior.

Sensitivity Studies

The previously established workload model allows the analyst to also investigate the sensitivity of the operator's workload and performance to variations in the driving environment. The analysis is focused on the highway driving environment, because the highway model captures the intended environment for most of the current driving automation technologies, and many of the resulting automation-based driving experimental studies. The sensitivity studies evaluate the impact of varying road conditions, specifically traffic flow rate and variations due to the stochastic layout of background elements. First, the current highway model assumes a singular, average traffic flow rate for comparison to the city environment; this simplifying assumption does not capture the realistic variations in traffic flow and other road conditions. A benefit of the virtual workload metric is the ability to quickly assess differences due to flow conditions, among other environment parameters which have not been the focus of this study (e.g. weather, speed limits, etc.). The second sensitivity study is focused on workload variations due to the stochastic nature of the current agent-based model. Under the same flow rate and road layouts, variations in the virtual workload are experienced based on changes to the initial position of background vehicles and the resulting behaviors. This sensitivity study attempts to virtually capture the variability experienced by participants in real-world environments (e.g. studies have shown that the number of lane changes taken by a driver during commutes ranges from 0.12 to 0.52 lane changes per mile [62]). Together these two sensitivity studies demonstrate the methodology's applicability to assessing operator workload in variant environments.

Starting with the effect of flow rate, the baseline of approximately 1,000 cars per hour per lane is compared with a higher and lower flow rate. The higher flow rate is 2,000 cars

per hour per lane, which is above the free flow rate of 1,300 but not grid lock traffic [16]. The virtual workload for the higher flow rate is shown in Figure 3.15a. Comparing these results to the baseline (Figure 3.8), the higher flow rate has an expected higher maximum 29 cars in memory, however, it is not double that of the baseline's 17 cars. The average mental demand and temporal demand show a similar behavior, although temporal demand is close to doubling. Figure 3.15b shows the workload for a reduced flow rate. The reduced flow rate is 250 cars per hour per lane, which is low for a highway and shows the extents of the environment. In this condition, the driver is often driving with no neighboring vehicles, and shows an average mental and temporal workload that is more than 4 times lower than the baseline. For ease of comparison, the workloads have been normalized utilizing the workloads from the baseline flow rate and are shown in Figure 3.15c. As expected, performance is inversely related to flow density, because the "driver" has more difficulty maintaining its preferred speed. With more cars on the road, the "driver" requires a higher average and temporal demand, however, the scaling is non-linear. Environmental changes show the dependence between workload and traffic density. This sensitivity study highlights the difficulty in aligning virtual and experimental results, or even two sets of experimental results. However, the overall promising alignment shown for the baseline case above and expected variations due to environmental changes also highlight the strength of the methodology. Workload can be assessed under a multitude of conditions without running a large number of on-road experiments, which are costly and unfeasible early in the design process.

The stochastic nature of agent-based modeling is utilized to capture the unpredictable nature of a driving environment. The second sensitivity study focuses on these variations, which are driven by changes to the environment and behaviors. The majority of this random behavior is captured in the background traffic. Although the number of cars in the model is specified such that the average flow is in the desired range, the speed and location of each car varies for each driving experiment. The primary behavioral changes are in the sensory model or due to changes in awareness. The different placement and movement

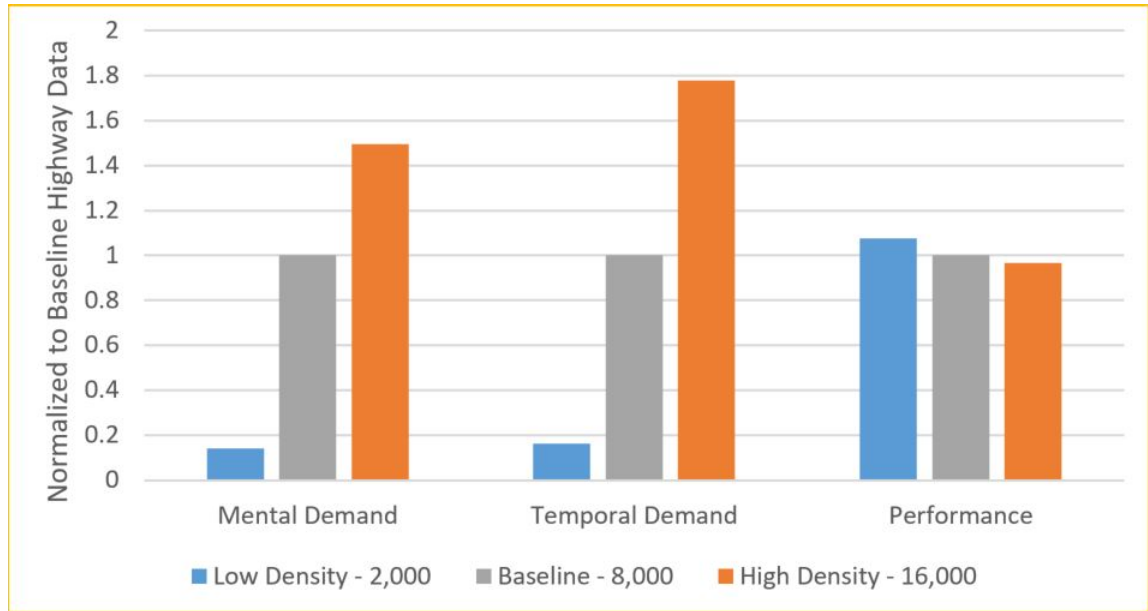
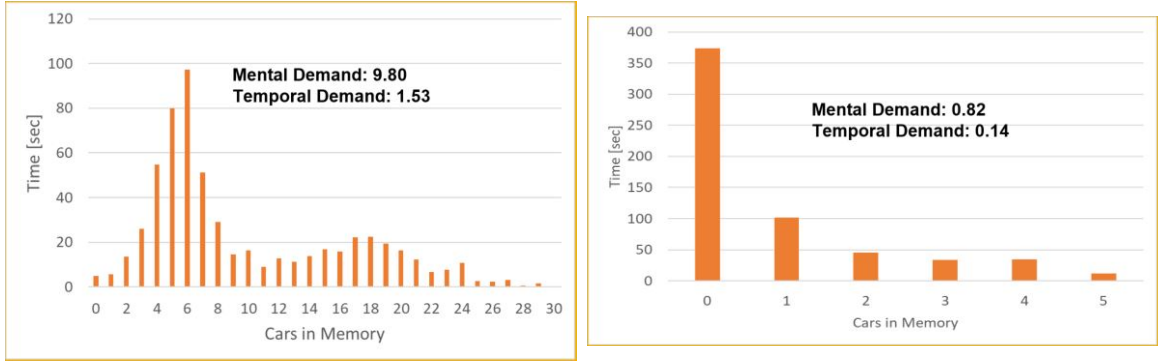


Figure 3.15: Workload variations for changes in the baseline highway flow rate

of background traffic affects the “driver’s” workload because more or less cars may be visible at any point. This new awareness effects the behaviors required (changing lanes or speed). Also, the time to acquire information from mirror and side checks is variant based on the defined distributions, thus the time to change lanes varies each time it is called. This random nature is controlled by a seed value, to allow the same experiment to be rerun exactly then the same seed value is utilized. Variations in the seed value result in different dynamic workload levels. Figure 3.16 shows the baseline highway run compared with two stochastic runs. This graph shows how each workload metric varies with each run. The

performance remained the most consistent between runs ranging from 89% to 92%, but showed notable temporal and mental demand variations based on the layout of the cars. Stochastic-2 has a similar average mental demand (6.57 cars in memory) to the baseline (6.56 cars in memory), however it has a maximum workload of 4 additional cars. Whereas, stochastic-1 has the same range for mental workload as the baseline, but a lower average (5.03 cars in memory) compared to the 6.56 for the other two cases.

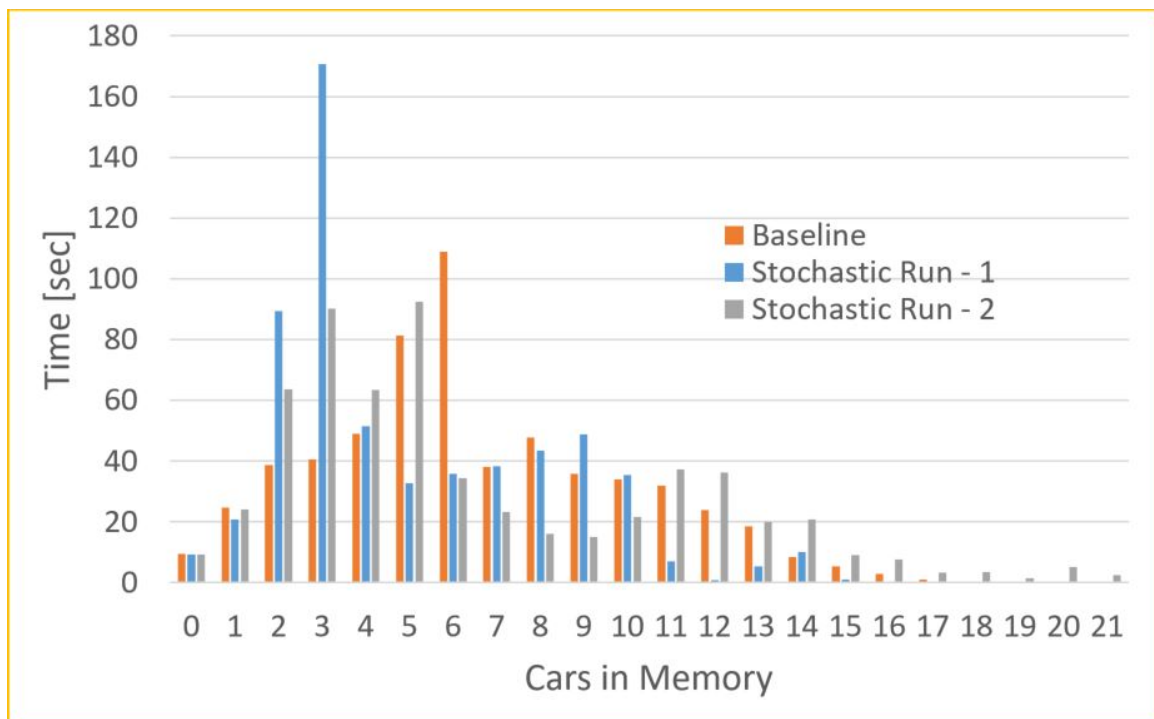


Figure 3.16: Mental demand variations for the highway baseline case and two stochastic runs. The stochastic-1 has an average mental demand of 5.03 cars in memory, average temporal demand of 0.72, and a performance of 89%. The stochastic-2 has an average mental demand of 6.57 cars in memory, average temporal demand of 0.88, and a performance of 89%.

To ensure that these variations are still in line with experimental data, Figure 3.16 adds these stochastic runs to the normalized comparison from Figure 3.14b. All three variations shows similar differences as described above. The mental demands are all similar, while temporal demand and performance show large differences. Since all three runs show acceptable comparison to the experimental data, the stochastic runs may be aggregated or looked at individually to provide workload insights. Aggregation allows the designer

to understand the extents of the operator’s workload on average. However, analyzing individual run data allows peaks, workload maximums and minimums, and duration to be examined. The capability to look at individual runs and understand workload throughout the run is a primary benefit for this new methodology. This technique can be utilized to identify workload outliers that may be a threat the system. For example, one run may show a long downtime created by automation integration and specific environment conditions (ex: stuck behind multiple slow vehicles). This condition may not be readily apparent and it may not exist under all traffic conditions, however, through the emergent behaviors and the metrics created by this methodology, areas of concerning workload may be identified when integrating automation technology.

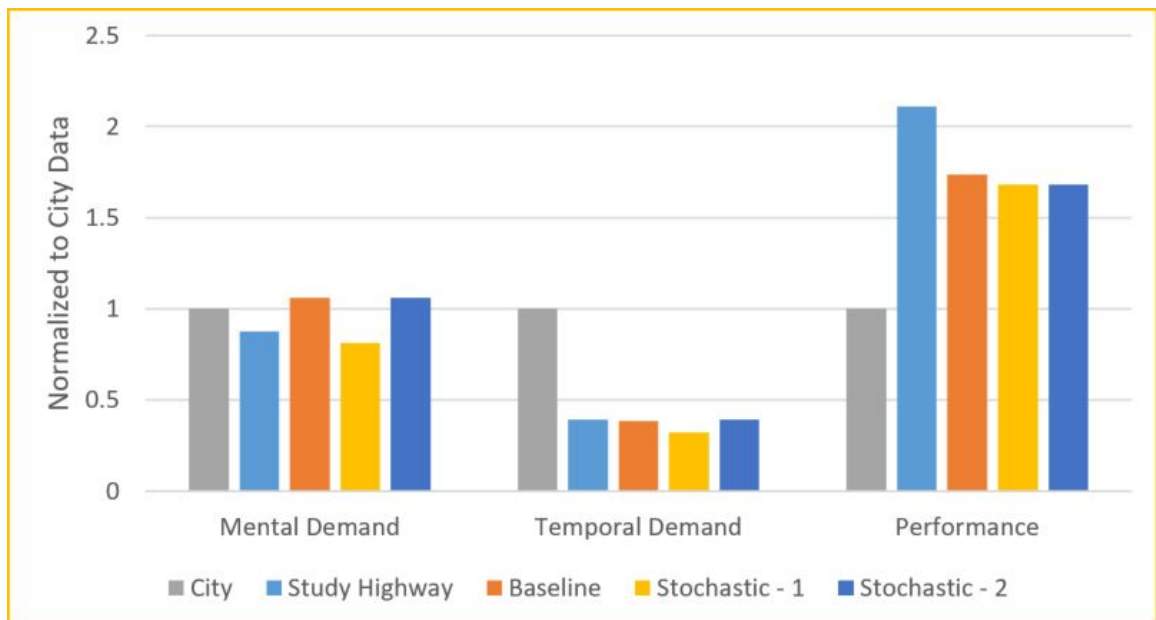


Figure 3.17: Normalized comparison between the human-based study done by Rahman et al.[86], the baseline highway case, and two stochastic variations of the baseline run. Normalization is done using the city workload data.

3.2 Experiment 2 - Automation Modifications

The reduction in the uncertainty surrounding operator workload is driven by the integration of automation technologies. Experiment 1 benchmarked the new methodology to existing

human-based experimental data to ensure the virtual metric could provide useful insights. However, this methodology must be able to capture operator behavioral and workload variations due to automation integration in order to satisfy the primary objective of the methodology. For technology integration, it is asserted and discussed above that it necessitates the addition of two new behavioral branches for the “driver”. The first branch is monitoring and covers how the “driver” interacts with the automation system. The second branch addition is correction and dictates new actions the “driver” takes to help the automation increase performance. With the addition of these branches, it must be determined how often the automation will be monitored. This correlates to operator trust and Hypothesis 2.2, “by changing the duration which data is stored in the operator’s SA model (if also monitored by the automation) and the frequency of monitoring tasks based on the level of trust in the technology, it will approximate the performance differences shown by different skill-level operators”. Operator trust and behavioral changes must be captured to accept Hypothesis 2.0 which states “the modifications, additional two branches on the dynamic task list and variations to SA according the operator trust, will allow technology trade space exploration based on operator workload and system performance”. For a refresher this connection between the research questions, hypotheses, and experiments refer to Figure 3.1.

Experiment 2 tests Hypothesis 2.0 and Hypothesis 2.2 while validating Assertion 2.1. These prescribed modifications are done based on the inclusion of two driving automation technologies: adaptive cruise control, and highway autopilot. For adaptive cruise control, the operator specifies a preferred speed and the car travels that speed unless impeded by a lead car, then the car slows to the lead car’s speed. Highway autopilot maintains the functionality of the adaptive cruise control, but is also capable of and will preference changing lanes if any are determined available. Both automation technologies are assessed in the highway environment, and the results are compared to experimental data.

3.2.1 Experiment 2 Setup

Experiment 2 focuses on the required changes to include automation technology, therefore the baseline environment from Experiment 1 remains unchanged. At the system-level, the “driver” behavior tree is changed for each automation technology. Additionally, behaviors are added to the system, separate from the driver, as the automation is able to direct actions within the environment. Therefore, the setup must be looked at separately for each automation technology. Starting with the simpler of the technologies, the adaptive cruise control (ACC) is currently being deployed by many automotive manufactures. ACC provides automated longitudinal control over the vehicle, setting the acceleration and braking to maintain a desired time-gap or preferred speed if no obstruction is detected [68]. The desired time-gap is assumed to be three seconds, which aligns with the assumption for manual control. The behavior tree for the “driver” utilizing ACC is shown in Figure 3.18. ACC requires a monitoring task of speedometer checks, and a correction task of a lane change if the ACC is hindering performance. Speedometer checks were previously untracked because decelerations were set based on closing distance to the lead car and accelerations were conducted when space was available, leading to on-demand checking. Lane changes occur when the “driver” feels impeded by the lead car. Previously, the driver would be aware of this impedance when he decelerates. However, ACC does not alarm the operator during deceleration. Therefore, the “driver” must have awareness that performance (distance traveled) is decreasing, and this is done through a speedometer check. The number of speedometer checks is determined based on the operator’s trust in ACC. Higher trust in the system leads to reduced checks, while low trust results in many speedometer checks. The frequency of checks is held as an independent variable, and assumed to have a log-normal distribution with a specified mean and a standard deviation of 33% of the mean. Every time the operator checks the speedometer, the car’s current speed is stored in the “driver’s” memory. This stored speed is used to determine if correction (a lane change) is considered. The steps and other considerations for a lane change are the same as manual

driving. ACC is capable of utilizing the far-forward sensor channel, but all other channels must be activated and checked by the operator. Additionally in the modified behavior tree, the speed control branch is deactivated from the “driver’s” behavior tree, because ACC is always assumed to be engaged. The steps for automation activation and deactivation are not currently considered.

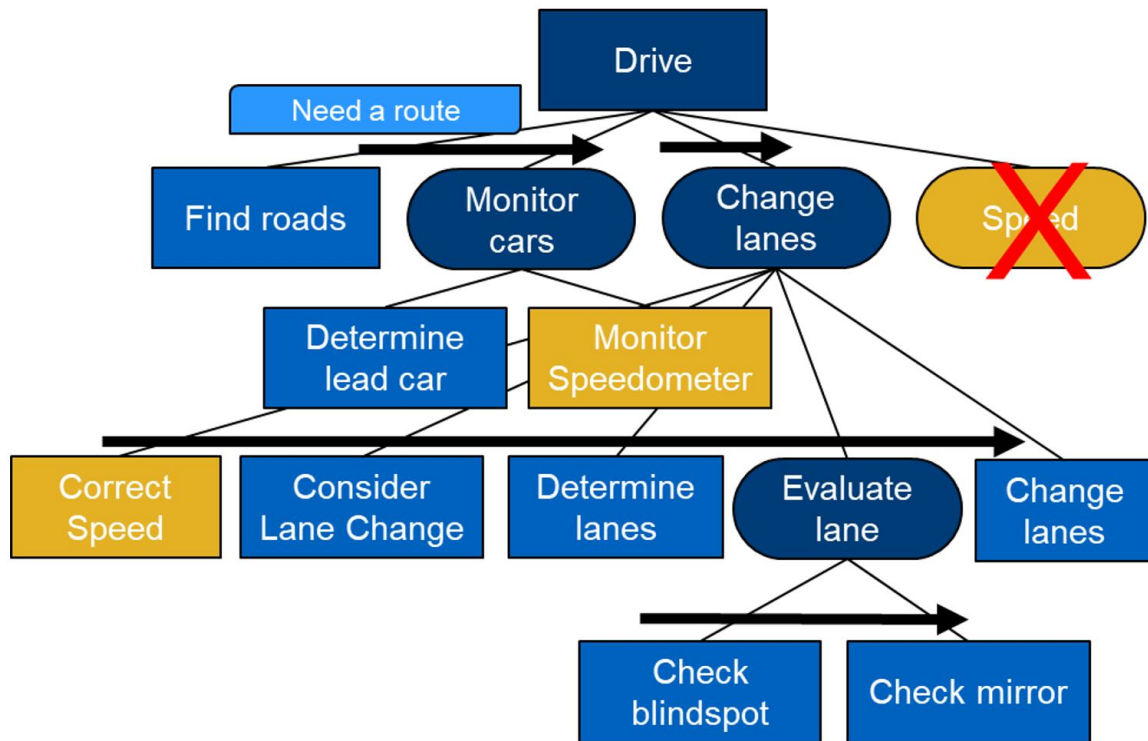


Figure 3.18: “Driver” behavior tree for adaptive cruise control (ACC). ACC monitoring has been added to the monitor branch with a speedometer check, and ACC correction is added to the change lanes action based on last remembered speed. The speed control branch is removed from the “driver” and controlled by the car.

While ACC provides longitudinal automation, highway autopilot automates both the longitudinal and lateral driving directions. Unlike ACC where the operator has an incentive to supervise the system and increase performance, autopilot requires the “driver” take a supervisory role instead of an active decider and planner. Similar systems have been built by many manufacturers but all of them function similarly, the “driver” is responsible for monitoring all aspects of the road, while the technology handles much of the workload [7]. The system’s sensory model must now be split between the operator and the automation.

The operator still must activate different awareness channels, but the highway autopilot is capable of monitoring all sensory channels. For automation, all six sensors are continually on and do not have the delay to check sides or mirrors.

Highway autopilot systems use an extensive set of sensors to provide information to the technology. For example, Tesla’s AutoPilot 2.5 uses eight cameras and twelve ultrasonic sensors to provide necessary road information [107]. In the current virtual model, the type and fidelity of each sensor is not analyzed thus the capability of the sensors is the focus. The sensor coverage for the operator from Experiment 1 is in line with that utilized by Tesla as shown in Figure 3.19. For the highway model, the far forward sensor channel is capable of seeing 160 meters, while Tesla’s main forward camera is focused for a max range of 150 meters [107]. Tesla’s wide forward camera has a maximum focused distance of 60 meters, compared to the 50 meters set for the virtual model’s near forward sensor [107]. Therefore, the desire to change lanes and speed will be done utilizing similar information. Both sensor suites also include side and mirror sensors to provide full coverage around the vehicle and decide availability of the lane. The same sensory capabilities are utilized for both the operator and the autopilot, however, the addition of the autopilot changes the system’s behaviors.

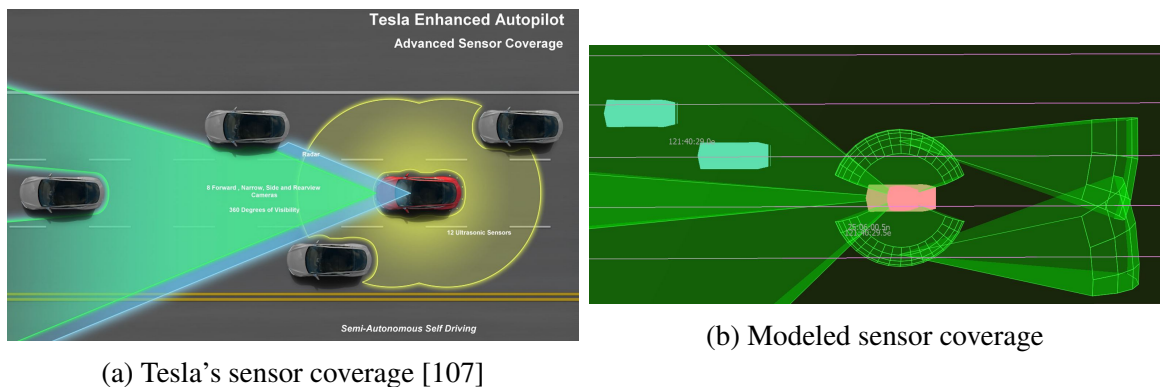


Figure 3.19: Depiction of sensor coverage currently utilized by a highway autopilot system compared with the sensor coverage modeled. The sensor coverage modeled matches the ranges utilized in Experiment 1 but also shows similar coverage to the autopilot

The addition of the autopilot moves both the “change lanes” (lateral control) and “speed”

(longitudinal control) branches from the “driver” to the automation technology. Lane changes are constantly analyzed based on current speed, the lead car’s speed (if applicable), and neighboring lanes. Unlike ACC, the driver does not activate lane changes, so actions are done based on the automation’s understanding of the environment. The steps to changing lanes remains the same to find an available lane that is going faster, however, the system no longer has delays to check blindspots or mirrors because the automation is always monitoring these regions. Additionally, speed decelerations, when required, utilize the lead car’s speed rather than rate of change for the separation distance since the automation is capable of measuring the lead car’s speed. Moving these behaviors to the automation removes the operator’s sensory cues to activate sides and mirrors channels. Therefore, the addition of the monitoring and correction branches is difficult. Since the operator is assumed to be monitoring the technology, it is assumed that the operator is aware of every car that the automation is tracking. Correction for the highway autopilot is typically operator intervention, however, this branch is not currently explored because it required too many assumptions regarding failure modes. Failure modes are a promising expansion of this methodology, but out of the current research’s scope.

3.2.2 Experiment 2 Results

Experiment 2 assesses the methodology’s ability to capture workload variations due to technology integration. To capture these variations, adaptive cruise control (ACC) and highway autopilot automation technologies are added to the environment from Experiment 1. ACC represents the inclusion of an SAE level 1 technology, while highway autopilot is either a SAE level 2 or level 3 technology depending on the source [107] [89]. Starting with ACC, the workload and operator actions throughout the simulation are shown in Figure 3.20. ACC changed the behavior tree to include monitoring and correction elements, while the speed tasks of acceleration and deceleration are moved to the automation. The frequency with which the operator checks the speedometer is related to their trust in the automation.

For this run, the mean of the lognormal distribution is assumed to be thirty seconds. The “driver” also updates his speed in memory through a speedometer check every time a lane change is conducted. This results in a total of 31 speed checks and 147 speed changes. In comparison, the baseline case shown in Figure 3.10 has 26 acceleration tasks and 55 deceleration tasks for a total of 81 speed change actions. Overall, ACC at this trust level (mean of 30 seconds) required fewer actions by the operator, and resulted in more frequent system speed changes as it continually updated to the lead car’s speed. The car utilizing ACC also conducted fewer lane changes at 8 when compared with the baseline’s 15 lane changes. This is partially attributed to the rapid lane changes in the baseline as it searched for a faster lane, however, this also shows an increased willingness by the “driver” to handle speed variations by the lead car due to a lack of speed awareness. ACC caused a decrease in average mental demand from 6.55 cars in memory for the baseline to 4.65 cars in memory when ACC is included. This decrease in mental demand is consistent with the ACC study conducted by Stanton & Young [101]. Temporal demand also decreased when ACC is included. Temporal demand for the benchmark case is 0.86 [1/sec] while it is 0.55 [1/sec] when ACC is included. The decreased lane changes decrease the variation in cars be added and removed from the “driver’s” memory. This is intuitive as fewer mirror and side checks are conducted. Lastly, the system’s performance slightly increased to 94% when ACC is included, compared with the 92% for the benchmark. This performance increase is due to the continual and exact speed variations maximizing distance traveled.

Comparatively, the addition of highway autopilot added lateral automation to the longitudinal control provided by ACC. This moves both the “change lanes” and “speed” branches from the “driver” to the automation technology. Therefore, highway autopilot reduces the modeled dynamic actions of the “driver” to be near zero, and highlights some of the challenges with highly-automated SAE level 2 technologies verging on level 3 technologies. The results from this study are shown in Figure 3.21. Unlike the other action and workload graphs, this one is focused on the awareness levels of the automation. Since the

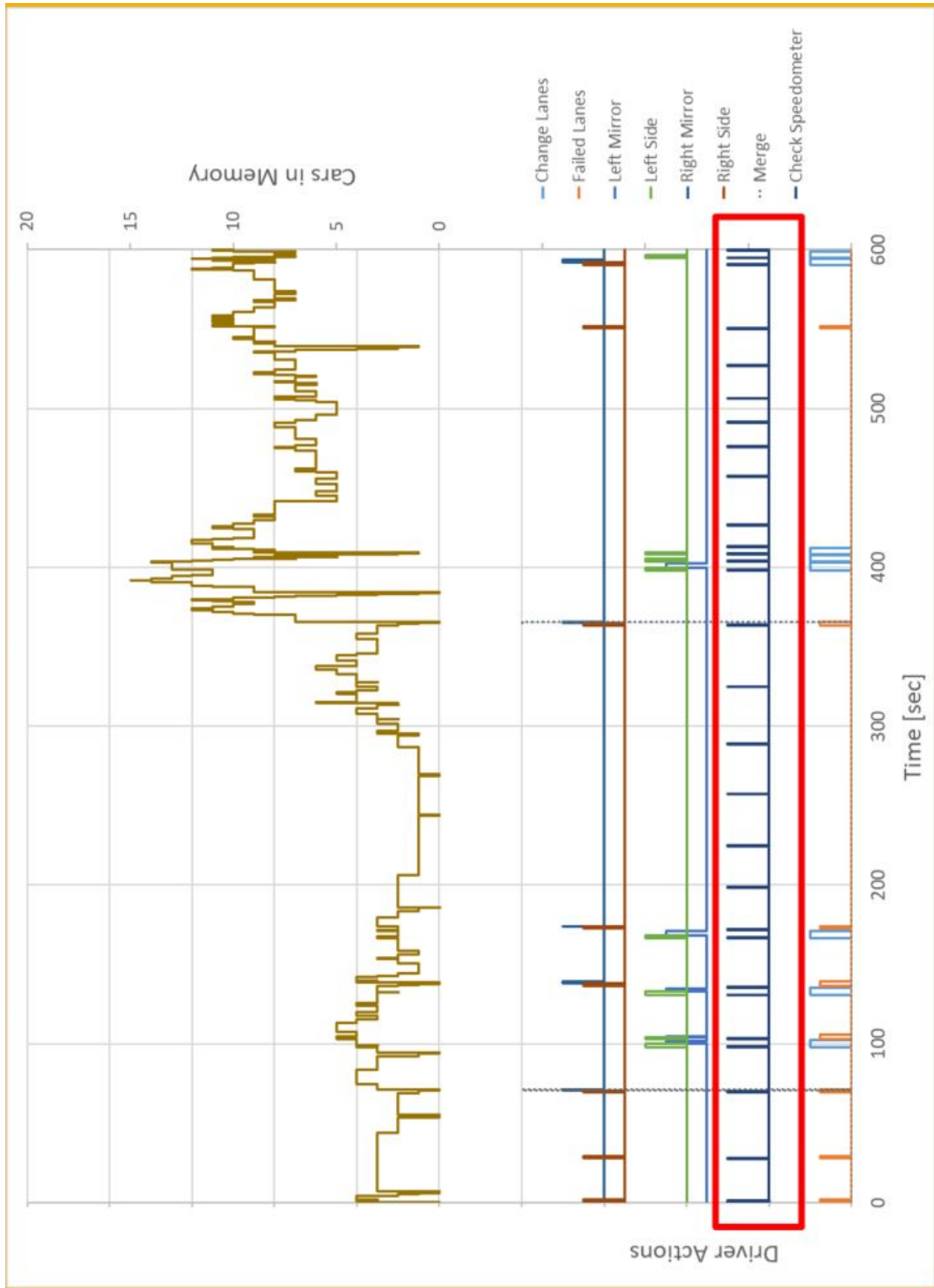


Figure 3.20: The actions taken by the operator while utilizing adaptive cruise control (ACC). ACC causes a replacement of speed increases and decreases with speedometer checks. The speedometer checks are based on a lognormal distribution with a mean of 30 seconds. The same baseline, 8,000 car scenario has been used. The width of the spikes demonstrates the duration of the action.

operator is generally responsible for the automation, it is important to understand what information is being tracked, decisions being made, and the duration of those decisions. The autopilot's mental workload requires 6.80 cars which is higher than the 6.55 cars required for the baseline and the 4.65 required with only ACC. It also requires a higher temporal load of 0.66 [1/sec] than ACC's 0.55 [1/sec]. However, this is lower than the baseline's temporal load of 0.86 [1/sec]. These changes compared with the baseline are driven by the automation's continuous monitoring of the mirror and side channels, creating a higher mental demand (cars in memory) but lower temporal demand (cars entering and leaving awareness). This increase in autopilot mental demand could be further increased when automation interaction tasks are added (how is the operator ensuring the automation is functioning). The autopilot showed more frequent lane changes than either the baseline or ACC as it continually attempts to find the fastest speed and does not require time gain awareness. The lack of time required to gain awareness highlights a possible hazard when including supervised automation, because the operator takes longer to gain awareness for the action than the automation. Currently, autopilot can execute the action before it is approved by the operator [109]. This disconnect leaves the operator expected to maintain a high level of awareness and mental workload without being required to conduct any actions. De Winter et al. [116] conducted a survey of automated driving studies and found that most studies have been conducted in simulators and highly automated driving led to an almost 50% reduction in overall NASA TLX workload, and highlighted that autopilot is starkly different from ACC because the operator has the opportunity to divert attention to secondary tasks. There are concerns that prolonged exposure to automated driving results in driver desensitization and overall task disengagement [7]. This disengagement is seen in practice with operator's sleeping or heavily distracted while 'monitoring' these highly automated system's. Discussed in the motivation section, the NTSB partially blamed Tesla for Autopilot design flaws allowing too much disengagement from the driving task leading to crashes [85]. With the inclusion of automation, it is important to understand the opera-

tor's workload but also by capturing the action list this methodology provides insight into the operator's task involvement. The operator's involvement often varies based on trust.

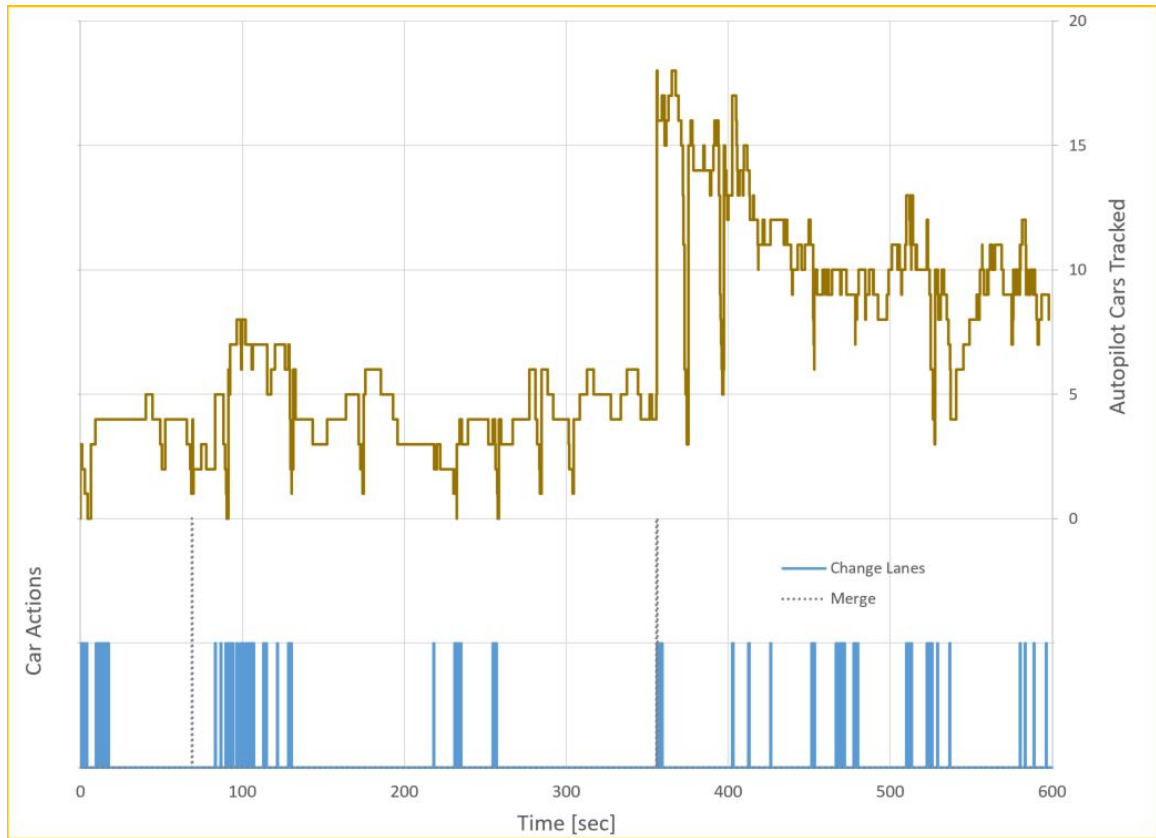


Figure 3.21: The actions taken by the automation while utilizing highway autopilot. The same baseline, 8,000 car scenario has been used. The width of the spikes demonstrates the duration of the action.

Trust and the level of task interaction have been shown to play a large component in the operator-automation interaction and resulting performance. Experienced operator's understand the limitations of technology, therefore they collect manual data more frequently to ensure that it is functioning properly [56]. This can be captured in the virtual environment by changing the operator's awareness of the automation's performance. In the highway environment, the operator's trust in ACC is captured by varying the frequency of speedometer checks. Checking the speedometer correlates to a manual collection of data in the longitudinal direction, which is currently controlled by ACC. More frequent speedometer checks demonstrates a lower level of trust in the automation, while less frequent checking corre-

lates to higher trust levels. The baseline highway environment is utilized for all cases. The only variation is to the dynamic task list and changes the mean (and corresponding standard deviation as outlined above) for the lognormal distribution controlling the operator's speedometer checks in the monitoring behavior. The results of the virtual trust study are shown in Figure 3.22. Three different mean speedometer checking frequencies are shown: 15 seconds, 30 seconds, and 45 seconds. Since each data category presented uses a different scale, they are normalized based on the 15 second case to show all three trends on the same graph. The 45 second case has 40% of the speedometer checks (or a 60% reduction in actions) when compared with the 15 second case. It is not exactly 33% because the times for speedometer checks are pulled from the lognormal distribution and it is checked every time that the operator finishes a lane change. The reduction in workload and higher trust in the automation results in lower overall system performance, as the operator becomes less aware of when the ACC is causing a performance reduction. This demonstrates a change in awareness as the operator becomes complacent with the automation's functionality. The correlation between performance and the operator's involvement through speedometer checks is in line with the gunnery performance findings from the Army Research Laboratory discussed in the motivation[13]. Similarly, the mental demand decreased as the operator's trust increased. With higher trust in the automation, the "driver" behaviors for activating side and mirror checks were less frequently utilized. The temporal demand increase shown between the 30 second case and 45 second case is unexpected. This increase corresponds to a higher pace of cars entering and leaving memory for higher trust levels. This is a small increase and may be explainable if the temporal demand changes are being driven by the environment, since the majority of time is spent looking forward especially as trust increases and the environment remains the same except stochastic variations. Both the 30 second and 45 second cases still showed a lower temporal demand than the 15 second case.

Experiment 2 tests hypotheses 2.0 and 2.2 by including automation technologies into

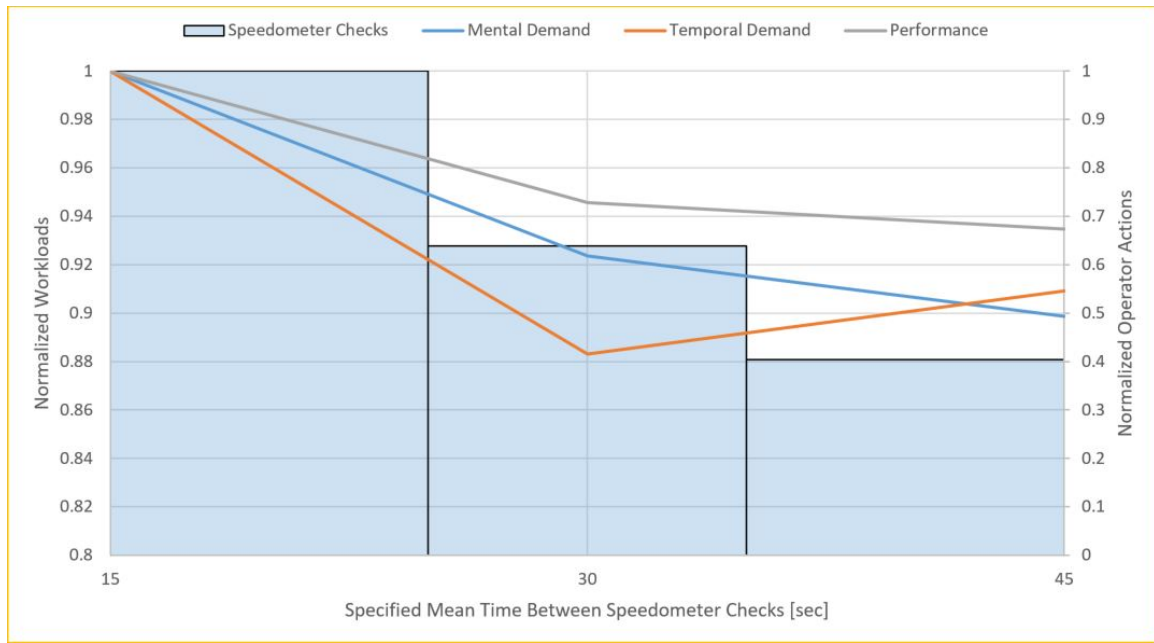


Figure 3.22: Normalized comparison of the operator's workload categories and actions with ACC integration. The specified mean time between speedometer checks captures the trust that the operator maintains in ACC (lower trust results in more frequent speedometer checks).

the framework benchmarked in Experiment 1. The inclusion of ACC and Highway Autopilot demonstrate the minor variations required in the sensory model and dynamic task list required to capture the teaming effect between the operator and different types of automation. ACC is further utilized to test the methodology's ability to capture operator trust (Hypothesis 2.2) by varying the frequency that the operator checks the automation. The baseline highway environment is utilized for all comparisons. The resulting comparisons between the virtual workloads and performances and experimental data found in literature showed promise. The Highway Autopilot showed the highest performance, however, it also required the highest level of workload for the operator to be monitoring the automation and did not involve the driver in any driving actions. ACC's performance is driven by the level of trust in the automation since it only controls the longitudinal direction. For the highest performance level's, the operator must check the automation frequently. This performance variation driven by changes in the time between speedometer checks validates

Hypothesis 2.2. Expert operators are likely to frequently check the speedometer, because they know how the ACC affects their performance, while novice operators may become more complacent and allow the ACC to control speed for longer intervals without checking the speedometer. The variations in the virtual results with the inclusion of technology and alignment with experimental findings leads to the acceptance of Hypothesis 2.0. The addition of monitoring and correction branches to the behavior tree changes the awareness and workload of the operator to capture the teaming effect. These changes allow decreased uncertainty surrounding the operator's actions while utilizing ACC and Highway Autopilot. The inclusion of multiple technologies and quantification of workload enables trade space exploration. However, the utilization of this new information to evaluate technologies must still be analyzed.

3.3 Experiment 3 - Automation Assessments

The new methodology seeks to inform requirements definition, therefore it is important to understand how the new metrics can be utilized in the early design process. As discussed in the motivation, the TIES process is utilized to introduce and select technologies, however, automation technologies lacked quantitative metrics for the evaluation step. The first two experiments show the new methodology's ability to explore an automation technology trade space and the quantification of the operator's cognitive load. This experiment tests Hypothesis 3.0 by utilizing those results to compare and evaluate automation technologies. Traditionally, NASA TLX provides a single workload data point for an experiment. However, the dynamic measurement of workloads and actions throughout the operational scenario is capable of providing more insight into the effects of automation integration. The trade space must be capable of comparing the operator's workloads and actions throughout the scenario. It should also be able to identify possible failure points due to workload peaks or long lulls.

3.3.1 Experiment 3 Setup

A trade space analysis examines alternative approaches of achieving a set of desired outcomes [78]. In the IPPD process, this correlates to evaluating alternatives (Figure 2.5) and in the OR process, a trade space falls in the output analysis and interpretation step (2.7). The trade space is determined by the problem at hand, with the problem definition and system requirements driving the evaluation metrics. They are multi-dimensional with typical criteria including performance, schedule, risk, and cost [122]. A notional trade space capturing cost, benefit, and risk is shown in Figure 3.23. Ten architectures are represented with the circle centers determined by the design's cost and expected benefit. The diameter of each circle represents each design's risk or uncertainty in the cost-benefit analysis. A complete trade space helps decision makers identify areas for further study, points of diminishing returns, and desirable architectures. Since the trade spaces are often multi-dimensional, they can show equal desirability of multiple architectures; these architectures are said to lie on the Pareto frontier. The Pareto frontier represents equally efficient architectures, where competing system objectives drives desirability of each design [67]. An example of Pareto optimal architectures, based on typical system objective functions, a high-performance, high-cost design compared with a moderate-performance, low-cost design. This concept is shown graphically on Figure 3.23. The goal of the decision maker is to maximize system benefit while minimizing system cost. Therefore, a design that is both cheaper and provides more benefit, such as Design 4 to Design 1, is considered better. However, designs that improve one criteria (decrease cost) while diminishing the other criteria (decrease benefit) are said to be Pareto optimal [67]. All designs design along the frontier have desirabilities that cannot be directly compared thus each design must be examined by the decision maker. Based on this trade space, further studies could be focused on improving the fidelity (risk) for the designs along the Pareto frontier or moving forward with a Pareto optimal set of designs.

Experiment 3 breaks down the operator-focused data to create a similar trade space

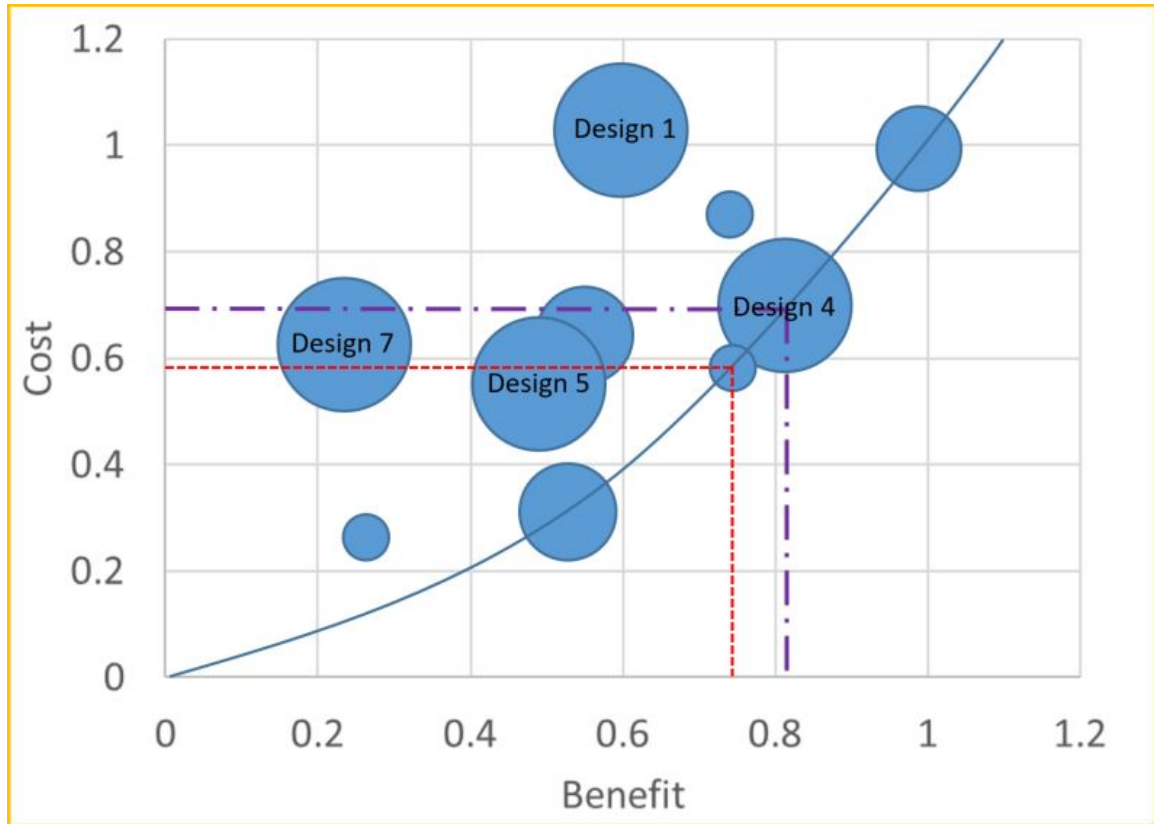


Figure 3.23: Example of a design trade space and Pareto frontier. Each circle is centered at an architecture's notional cost-benefit with the diameter representing design risk (uncertainty in the cost-benefit analysis).

for automation technology. This methodology is driven by the uncertainty created during automation introduction. Currently many OR studies capture the benefits of automation while creating high-levels of uncertainty between the automation and operator thus sometimes failing to capture the entirety of the trade. The small, example trade space consists of four automation levels/designs: the manual driving baseline, adaptive cruise control (ACC) with high trust, ACC with low trust, and Highway Autopilot. High and low trust for the same technology (ACC) are separated to demonstrate the difference in benefit and cost for different operator trust-levels. This skill-level difference in the cost-benefit is a trait not typically required and in this case is assumed to be separate from design risk. Design risk captures the technical, cost and schedule risk of selecting an architecture [122]. This risk is still present for automation technologies. Unlike risk which is at the system-level, the

trust-level is operator dependent and specifies how the team will function. It can change naturally with further interactions between the operator and automation or with further training. Therefore, it is important to understand how the cost-benefit of a system may change throughout its service life. Based on the literature review addressed above (ATC and ARL studies, Tesla failures, Boeing 737 MAX issues), three primary categories have been identified for automation technology comparisons: measured workload, operator actions, and awareness concerns.

The measured workload category contains the mental demand, temporal demand, and performance of the operator. These metrics were benchmarked versus literature results, and provide the first insights into automation. The performance is now a function of operator's awareness and environment conditions. The desire is to maximize performance and minimize temporal and mental workloads. The minimization of workload is intended to reduce operator stress and can free them up for other tasks. However, this change also impacts the operator's actions, and these actions should be analyzed to avoid unintended consequences. The second category captures the frequency of the operator's actions. These actions can be anything of interest from the operator-focused behavior tree. For the purposes of this analysis, the actions chosen are mirror checks, side checks, and number of lane changes. The frequency of required mirror and side checks helps provide indication of the involvement of the operator in system awareness. The number of lane changes is also tracked between the technologies. Frequency can stand in for proficiency, because skill deteriorates without practice (as discussed in concerns over aircraft's autopilot [117]). Combining required dynamic workload and action data provides insights to the last data category of awareness concerns, such as time not looking forward and time between lane changes. These pieces of data can highlight issues like the lack of operator involvement in the 2008 Tesla crash (Figure 2.35). Excess actions by the automation may be too difficult for the operator to supervise or lead to increased stress. Of similar concern, as automation takes over for the operator, the operator has capacity to increase task-irrelevant actions (checking phone, ad-

justing radio, etc.) and is more likely to start day-dreaming [122]. Similar concerns were highlighted by de Winter et al., “the driver of a highly automated car has the possibility, for better or worse, to divert attention to secondary tasks” [116]. This mind-wandering behavior for highly automated driving is shown in Figure 3.24. Figure 3.24a shows mind wandering frequency as a function of time. This depicts an increased likelihood of mind wandering as the driver remains responsible for a highly automated task. The issue of mind wandering is more prevalent at lower workloads as shown in Figure 3.24b. The linear fit shows increased time spent mind-wandering at lower workloads than when an operator has higher task loads. The insights from this study and the failures seen in highly automated systems motivate the awareness concerns in the trade space.

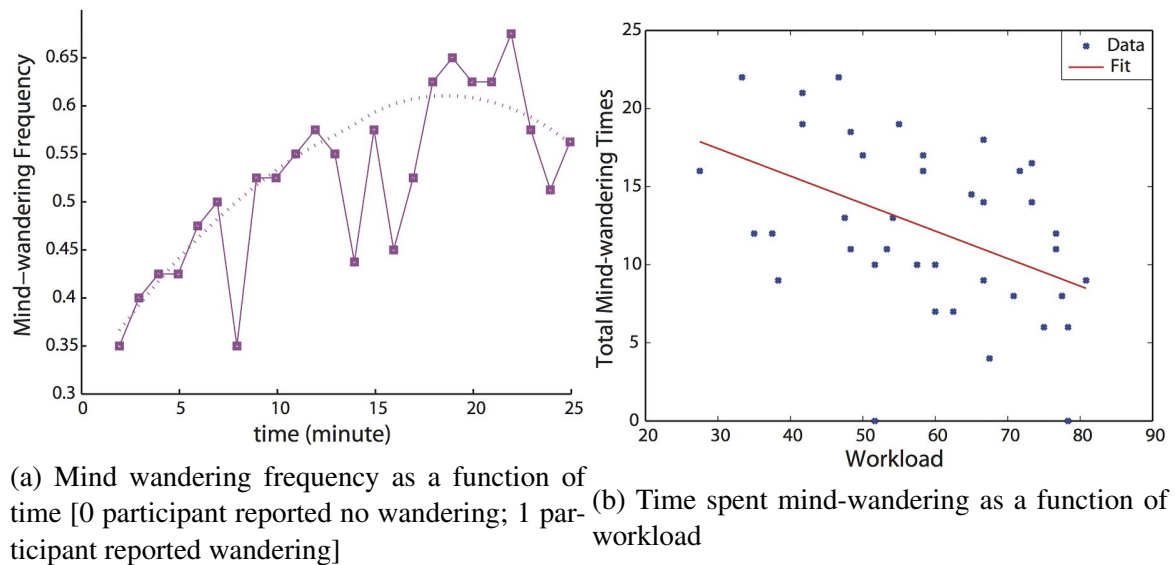


Figure 3.24: The operator’s mind-wandering is impacted by variations in workload and time performing the same task [122]

Overall, a trade space for automation technology includes three new categories: measured workload, operator actions, and awareness concerns. These have been determined based on the type of issues seen in literature being caused by automation uncertainty. The multi-dimensions of the trade space should show the benefits and limitations of automated driving technologies. For example, highly automated driving technology may out perform the operator, however, it leaves the operator monitoring and not involved thus likely for

mind-wandering. A similar study was discussed in the background, showing decreased performance for ATCs [27]. Experiment 3 creates a small trade space, but demonstrates the capabilities enabled by this new operator-focused methodology and tests Hypothesis 3.0.

3.3.2 Experiment 3 Results

Utilizing the highway environments from Experiments 1 and 2, workload, action, and awareness data was collected for non-automated driving, adaptive cruise control (ACC) at two trust levels, and highway autopilot. A scatterplot matrix summarizing these results is shown in Figure 3.25. Along the columns are the different technologies with each column's x-axis varying based on number of cars on the highway network. The trend lines for traffic density help capture the non-linear scaling of some attributes and provides better transparency to a decision maker compared with a single average metric or metrics from a single environment. The top three rows of the matrix (mental workload, temporal workload, and performance) show the workload category metrics metrics. Starting with performance, the highway autopilot shows the overall highest performance with variations in performance being less dependent on traffic density. ACC with high trust showed the greatest correlation between performance and density. Therefore, highway autopilot seems the most desirable, however, concerns start to arise as mental workload is examined. The third row shows three lines for mental workload. The top trend line (green) in each column represents the maximum number of cars in memory for each condition and design. The middle trend line represents a maximum one minute average for mental workload (the level of workload experienced by the operator for the most strenuous minute of the environment). The bottom trend line (orange) captures the overall average mental workload. These variations show the benefit of the dynamic capability of this new methodology. The average workload in many cases was half of the workload required during the most strenuous minute. As technology is chosen, this workload overlap can be better assessed and

designed for to ensure satisfactory mental workload throughout the environment. Overall, mental workload for ACC has a weaker correlation to traffic density than no automation or the highway autopilot. ACC allows the operator to focus attention to lateral control, while longitudinal control is automated. Highway autopilot automates both axes, but since the operator is supervising the automation it is assumed that the operator must have a workload similar to the automation. This raises an automation design consideration because the highway autopilot is intended to reduce operator workload and stress, but the risk and uncertainty is high since it requires a higher workload. The operator is unlikely to maintain this workload because he is disengaged from the task, as shown by the looking at the operator actions.

The fifth and seventh rows in Figure 3.25 shows the trend lines for operator actions. The fifth row shows the number of side and mirror checks taken by the operator for each environment and design. The green trend line (typically on top) shows the number of side checks, while the blue trend line shows the number of mirror checks. The number of checks shows the frequency (and proficiency) of the operator performing each action. The action data shows the an operator with no automation is likely to be the most familiar with awareness checks, followed by ACC with low trust, ACC with high trust, and finally highway autopilot. For highway autopilot (since it is automated both laterally and longitudinally) the operator is never required to check mirrors or sides. So although, he may choose to collect awareness, his action set does not require it and he may not be as diligent about gathering it [27]. This decreased frequency may drive requirements for increased training to ensure the operator remains proficient with the skill, or the automation design must force the operator to remain involved in the awareness task. A similar frequency analysis can be conducted when looking at the number of lane changes (Figure 3.25's seventh row). The autopilot conducts substantially more lane changes than any other case. This can be varied in the automation behaviors based on how lane changes should be decided, but currently the autopilot continually varies lanes as traffic around it changes speeds in an attempt to

maximize performance. The next most frequent lane changes occur without automation because ACC removes some operator awareness. All technology levels still conduct lanes changes, and at a pace that would be unlikely to leave the operator unskilled. The minimum is 3 lane changes in the 10 minute scenario for ACC with high trust. However, the pace of actions is also important to ensure the operator is not overloaded or underloaded.

The fourth and seventh rows capture possible awareness concerns. Continuing with the discussion on lane changes, the seventh row shows the time between lane changes. The top trend line (green) shows the maximum time between lane changes in the 10 minute scenario, while the bottom trend line (pink) shows the average time between lane changes. The time between actions can highlight possible issues with automation implementation. In this case, the average time for lane changes with no automation is 40 seconds with 8,000 cars compared with 10 seconds for the automation. The current highway autopilot implementation requires four times the lane change supervision as actually executing the lane changes. A secondary challenge is captured by the maximum time between lane changes. ACC's lower operator awareness increases the maximum time between lane changes, more noticeably with high trust. The environmental model enables the capture of outlier cases which may have high times between actions. These cases are significant because of the likelihood of operator mind-wandering. With no automation, the driver remains vigilant, however, as actions become automated, the operator allows the technology to takeover as they become disengaged [116, 122]. This disengagement can lead to lower performance and possible catastrophic failure if the automation fails without operator awareness. A second possible awareness concern correlates to the amount of time the operator may be focused or not focused on an action. An example of this type of data is shown in the fourth row capturing the amount of time that the driver is not looking forward. Any time spent checking mirrors or sides is considered time not looking forward. This time is natural in non-automated driving and was used during the benchmarking of the virtual model. The ACC with low trust and no automation driving showed similar amount of time not looking

forward. ACC with high trust decreased the amount of time not looking forward, while highway autopilot always allowed the operator to look forward. Increasing the time looking forward increases longitudinal awareness at the expense of lateral awareness. A lack of longitudinal awareness has created safety devices such as emergency braking, however, to properly supervise lane changes the driving should also maintain a level of lateral awareness. The insights into time spent on an action or away from an action can ensure proper and intended utilization.

As discussed, trade spaces can take a number of forms, are typically multi-dimensional, and developed based on system requirements [122]. The new data provided by this methodology provides insights into multiple different categories. However, the multi-dimensional trade spaces can be difficult to visualize. Therefore, a couple sample trade spaces are provided. Figure 3.26 shows the trade space between performance and mental demand for different technologies. Performance is the benefit of the system and should be maximized, while mental demand is the cost to the operator so it is trying to be minimized. This graph shows the increase in performance created by the highway autopilot, but also the higher mental workload required for supervision. ACC reduced the operator's required mental workload, but decreased the performance. The performance decrease is larger for high trust operators, typically novices who do not understand how the technology works thus do not monitor the automation as thoroughly. With more experience the operator's performance will increase towards that of manual driving but requires more monitoring. A second trade dimension is shown in Figure 3.26. This trade space again shows the system performance but now versus the average time between lane changes which should be maximized. Similar to mental demand, the goal is to minimize operator workload, and since lane changes require involvement from the operator it is desirable to increase the time between lane changes. Since performance is still in the trade space, the order of the technologies remains the same, however, the spacing has changed. ACC has a more noticeable impact on the cost to the operator. For mental demand, ACC maximum decrease in average mental

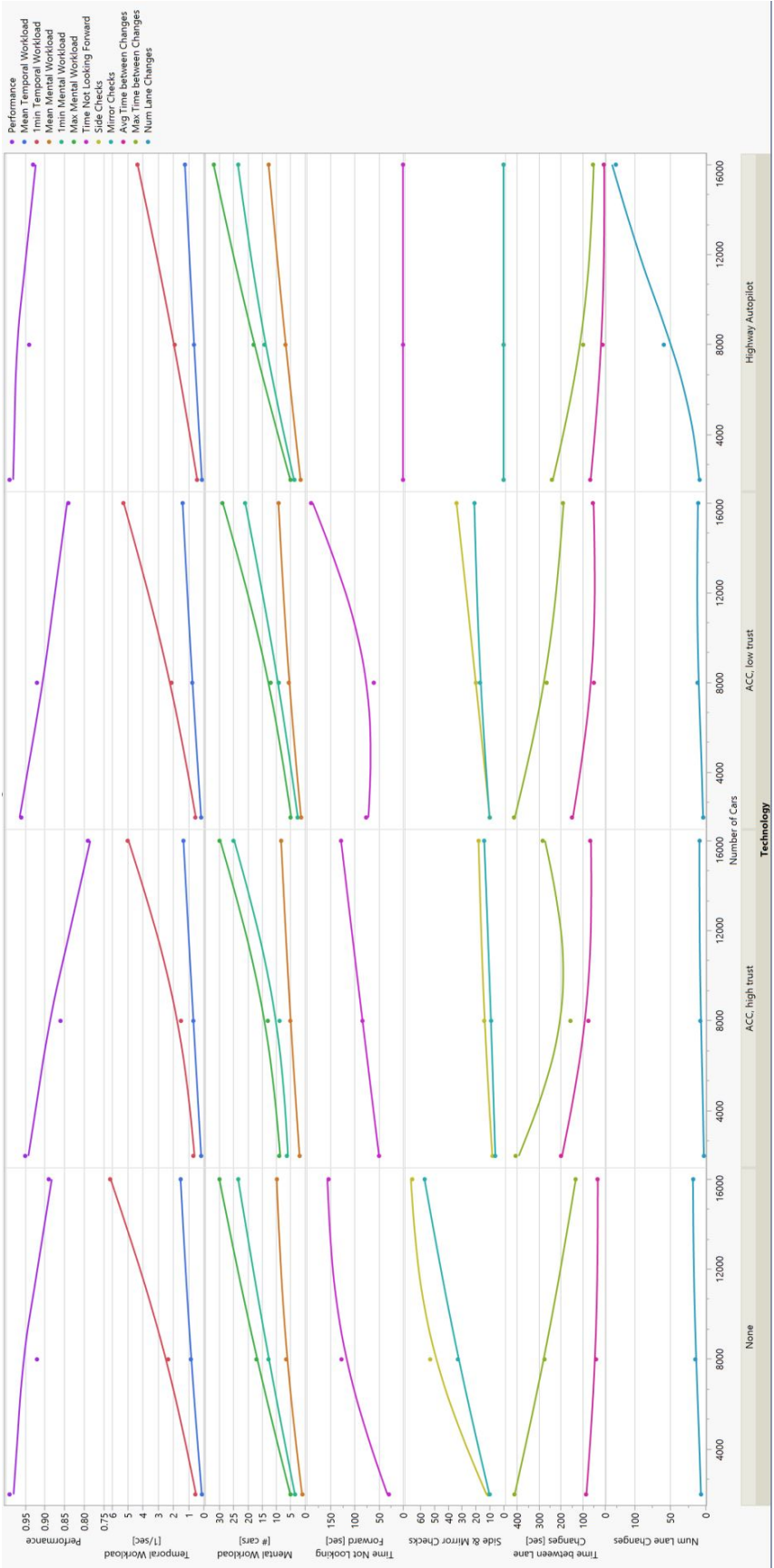


Figure 3.25: Scatterplot matrix summarizing the workload, action, and awareness results

demand is approximately 12% when compared with no automation while the average time between lane changes doubles for ACC. Highway autopilot increases performance but decreases the time between lane changes. Although the operator is not orchestrating each lane change, the desire to have them supervise increases the expected workload. Not shown in this trade space is the probability of mind wandering, but since the operator is not required in the action step and the performance remains the same, it is likely that they will perform secondary non-task related actions instead of maintaining awareness.

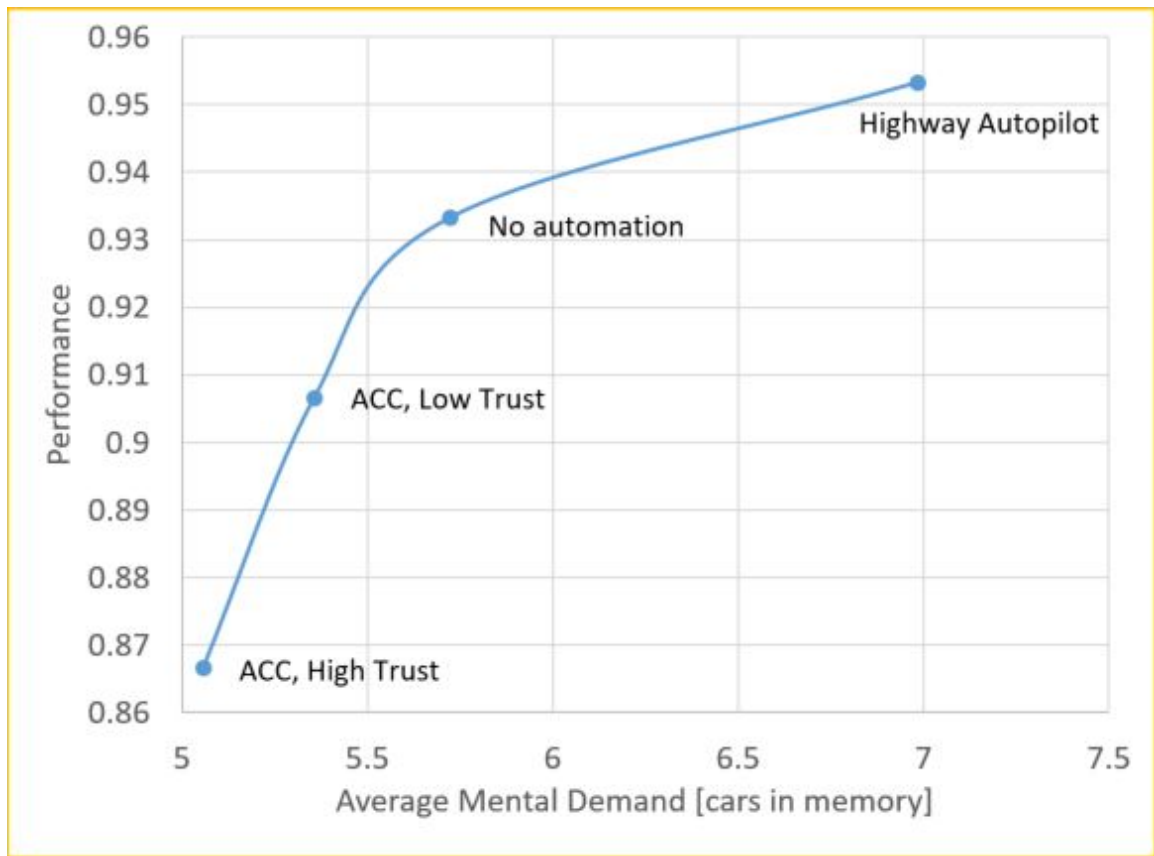


Figure 3.26: Example driving automation technology trade space showing the dimension of mental demand versus performance. Mental demand should be minimized, while performance should be maximized.

The new dimensions to the trade space provides the insights necessary for assessing automation technology. This operator-focused methodology for creating operations models enables three new metric categories (measured workload, operator actions, and awareness concerns). The new metrics are capable of capturing both the benefits and costs of

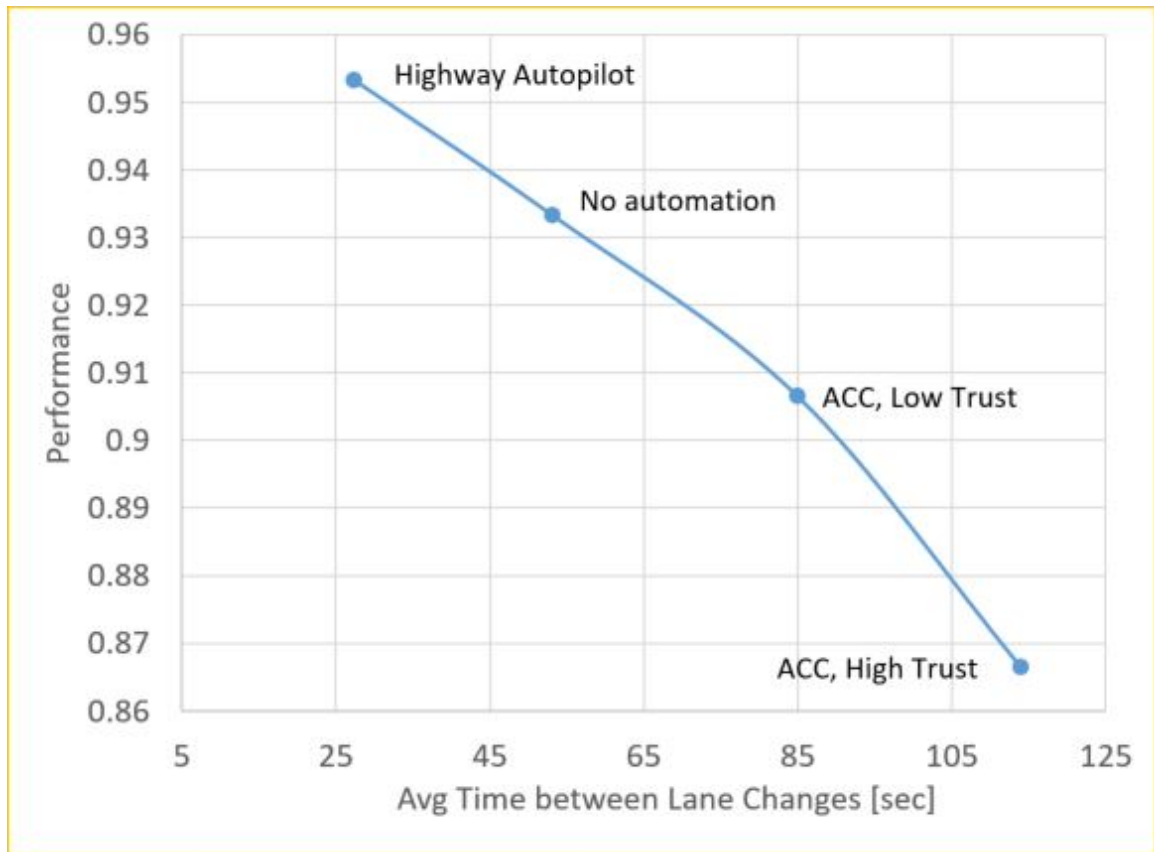


Figure 3.27: Example driving automation technology trade space showing the dimension of mental demand versus performance. Mental demand should be minimized, while performance should be maximized.

adding automation technology. Performance increases correlated to higher mental workload requirements, unless the operator is taken out of supervisor role, while reductions in awareness decreased team performance. Similarly, the lack of required interaction by the operator for highway autopilot is highlighted. The scatterplot results matrix shows no side checks or mirror checks even though the technology more frequently changed lanes. This leads to a high likelihood of mind wandering as seen in the Tesla awareness timeline shown in Figure 1.7. These new metrics and the insights revealed by comparing the results from the different automation technologies lead to the acceptance of Hypothesis 3.0. The new methodology provides metrics which allow decision makers to identify possible failure points and compares automation technologies. Capturing these possible issues early in the design process may allow more explicit breakdown between team functions and re-

evaluation or the selection of a different set of technologies. This limited design space shows the capabilities of this new methodology, however, it is only the starting point as it has only been applied to one environment.

CHAPTER 4

DEMONSTRATION OF METHODOLOGY TO AEROSPACE APPLICATION

Experiment 4 demonstrates the applicability of the methodology to the aerospace industry. The other experiments focused on the automotive industry due to data limitations for initial methodology verification, however, the goal of this experiment is to test Hypothesis 4.0 which states that this operator-focused methodology is capable of decreasing the uncertainty related to the early-design operational assessments of manned-unmanned teaming in military aircraft. The application is chosen due to the current trend towards manned-unmanned automation integration with technologies such as the Boeing Airpower Teaming System (part of the unmanned combat air vehicle (UCAV) program) [18]. The overall experiment shows how the new methodology can be applied to teaming systems and the insights it is capable of providing, which helps validate the methodology within the aerospace domain.

Although the military domain motivates this application, it is also a difficult environment to capture. The Airpower Teaming System is intended to increase the capabilities of manned fighter aircraft and act as a force multiplier [18]. However, unlike the driving scenario, the operational environment for fighter aircraft and the Airpower team is more variant and less homogeneous. There are many mission types currently performed by multi-role fighter aircraft ranging from air-to-air, air-to-surface, and reconnaissance missions, each having different flight profiles and action sequences for the pilot [33, 76]. A full system analysis assesses each mission type. However, for this experiment, only one mission type is utilized to demonstrate the methodology's utility. Since the Airpower Teaming system is currently in development, there is limited information about the system. However, pilot perceptions found in literature discuss that early implementations are not likely to be utilized in dogfighting or close-air support, instead their operations should be focused on intelligence,

surveillance and reconnaissance (ISR) and suppression of enemy air defense (SEAD) missions; “even a mission to strike a pre-planned target was deemed too complicated for a UAV to perform autonomously due to the uncertainties that may arise between the time the UAV is launched and the time it arrives in the target area, as well as the uncertainty surrounding positive target identification” [79]. Current UAVs, such as the Predator and Reaper, are designed for high-altitude, long-endurance in relatively uncontested airspace, however, future unmanned systems like theUCAV must be capable of loitering and attacking within heavily defended airspace, which requires systems with low-observability [41].

This assessment focuses on a fighter aircraft’s ability to perform a simple ISR mission, both individually and with an unmanned wingman, in contested airspace. The ISR mission is complete when the team reaches and captures data from three checkpoints then returns back to the rally point. When the pilot is on his own, he must fly each checkpoint, however, he will task theUCAV to checkpoints when available and loiter close enough to maintain communications. Whenever theUCAV reaches a checkpoint, it loiters until the pilot can ensure that the necessary data is collected before it proceeds. Throughout the mission, all of the aircraft attempt to avoid radar detection. A pair of red fighters fly through the scenario near the halfway point and engage the manned aircraft to capture a pilot distraction and variations in workloads, which may exist in a contested airspace. Similar to the car example, the aircraft’s behavior are built with a pilot-focus to provide insights into the instantaneous workload, and performance is specified as the time to complete the mission. The tradespace shows the performance and workload differences for the pilot with and without theUCAV.

4.1 Experiment 4 Setup

An independent fighter aircraft and a manned-unmanned team are modeled performing an ISR mission in a contested environment. The agents are modeled using the same four-component methodology as above for the driving trade study. First, the operations model

provides the environmental context and layout of all agents. The dynamic task list creates an operator-focused behavior tree to control the actions of the focus agent (the fighter aircraft). Linked to the task list, the sensory model designates when information is received from the environment and how it is utilized by the operator. Lastly, the workload model translates these operator's actions to a dynamic, quantifiable metric. AFSIM is again utilized as the agent-based modeling environment due to availability and applicability to DoD applications, however, the four-components can be implemented in any modeling language. The aerospace domain, especially the military side, provides some challenges that were not encountered in the traffic model. The information publicly available is limited and the inclusion of higher-levels of automation is more recent. This experiment shows how even limited amounts of data provides insights necessary for automation assessments, which can be improved with further data inclusion. Experiment 4 demonstrates the entire methodology so combines the setup elements from Experiment 1 and 2. Each of the four components are discussed in detail to create the initial model similar to Experiment 1, then the variations required for technology integration are described similar to Experiment 2.

4.1.1 Operations Model

The operations model provides the context for the ISR mission and the contested environment. The elements of the model are separated into red and blue forces, with the blue forces being the focus of the new methodology. A top-down view of the modeled environment is shown in Figure 4.1. A full ISR mission's flight profile includes segments for warm-up/takeoff, climb, cruise, descent, penetration, withdrawal, climb, cruise, descent, and reserves [76]. However, this has been simplified to focus on the technology in the battlespace. The blue force consists of a solo blue fighter shown in the bottom right of Figure 4.1. This aircraft is trying to achieve a surveillance objective at three locations throughout the environment. The objective is assumed complete at each checkpoint when the pilot ensures that the proper location has been achieved. The blue forces start from a rally point

(e.g. a tanker) that is flying off screen to the bottom-right corner then progress into the mission. This is figured to align with the cruise segment of the mission profile. When the fighters return to the rally point, the mission is complete. The blue forces ingress to Objective 1, proceed to Objective 2, then finish with Objective 3. The objectives have been placed just outside of the contested airspace to enable navigation between the three points. Figure 4.1 shows the equilateral triangle created by the three checkpoints with a spacing of approximately 170 nautical miles. There is a similar 170 nautical mile distance between the rally point and Objective 1. Overall this represents a flight path of approximately 850 nautical miles (the distance between Objective 3 and the rally point is approximately twice the spacing distance). The 850 miles is about half the range typically reported for fighter aircraft. The detailed routing between these objectives and deviations from the path are determined by the dynamic task list. All aircraft are point masses, with basic controls on how they operate in the environment. The aircraft are flying at a speed of 600 nautical miles per hour throughout the scenario, and are capable of making 6g turns. The blue aircraft carries four generalized medium-range, guided missiles with a specified range of 50 nautical miles. For simplicity, this model primarily focuses on two-dimensions, ignoring effects from altitude except line of sight considerations for communications and radar detection purposes.

The red forces consist of two radars and two fighters. The radars are static and the location are assumed to be known by the incoming blue forces. Each radar is assumed to have a 100 mile detection range for the piloted aircraft, which is between the ranges used commercially for air route surveillance (max 250 nautical miles) and airport surveillance (max 60 miles)[4, 61]. One radar is located in the center of the equilateral triangle, while the second radar blocks the direct path between Objective 2 and Objective 3. This overlapping nature embodies the intention and typical layout of integrated radar networks. The radar detection is not further modeled, because the blue aircraft are intended to avoid the two-dimensional radar footprint surrounding each radar. The red fighters are included to provide

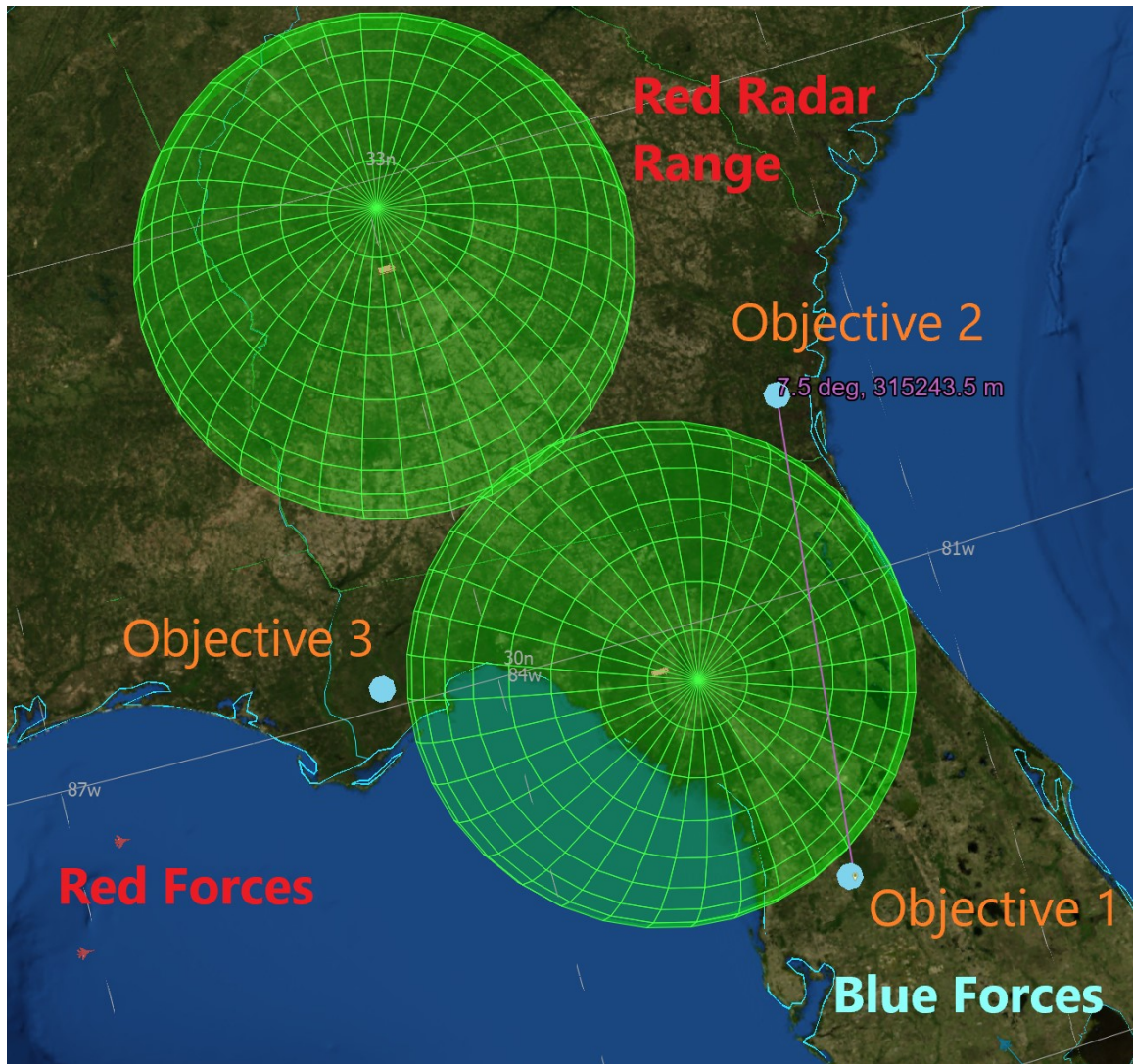


Figure 4.1: Operations model in AFSIM showing the layout of the red and blue forces.

a dynamic aspect to the contested zone. They start at the left side of the engagement zone and fly through the center of the blue objective space until they detect the blue aircraft. When the blue aircraft is detected, the red fighter turns to intercept. The red aircraft also utilize a geometric detector with a 100 mile range, but since they are moving through the space and their location is unknown the blue aircraft cannot avoid them. The air-to-air, or air interdiction, component of the scenario is not the focus of this scenario so the red fighter aircraft evasion behaviors are not captured. This is a basic representation but still is capable of capturing the pilot requirements as he attempts to navigate around a dynamic, contested

airspace.

4.1.2 Dynamic Task List

The dynamic task list is one of the three new elements required by the methodology to capture the operator's workload. It creates an operator focused behavior tree to control the actions of the agent in the operations model. A hierarchical task analysis (HTA) breaks down operator actions based on goals and sub-goals to translate to the operator's actions. For the task of driving in a city or highway environment, there is a high prevalence of publicly available data which can be utilized to develop a behavior tree, and where gaps exist personal experience in the driving activity can be used to help capture the hierarchy. However, the aerospace domain presents new complexities due to data availability limitations concerning the execution of ISR and other military mission types. Lacabanne et al. [60] present an HTA utilized by all pilots in flight (fighter, airliner, light plane) highlighting four meta-tasks: navigate, aviate, communicate, and system monitoring. The meta-task "navigate" controls where the plane flies, while "aviate" represents the pilot's control over the flight path. "Communicate" is data sent or received to external entities, and "system monitoring" is the actions required by the pilot to monitor and correct flight systems (ex: hydraulic system). The Lacabanne et al. HTA shows overall aircraft goals, while Endsley's work shows an HTA for engaging enemy aircraft [60, 30]. Endsley's HTA specifies the top-level goals as: kill enemy aircraft, do not get killed, avoid detection by the enemy, reach point X with Y weapons with Z fuel by time W, defend space X, and defend aircraft [60]. Using these HTAs as a guide, a modified version more applicable to this scenario and software agent creation is shown in Figure 4.2. The navigate meta-task is broken into "assess threat level" and "monitor navigation", while the aviate meta-task constitutes "protect aircraft", "correct course", "capture recon target", and "return to rally point". Breaking apart the goals, better aligns to the sensor model because it can show when information may not be acquired. For example, navigation monitoring is unlikely to acquire when the

pilot is focused on protecting the aircraft. Therefore, the sensor model should not acquire new location and heading information, and the route will need to be re-planned after the engagement. For simplicity, the communicate and system monitoring meta-tasks are not captured. System monitoring, awareness, and correction is the most relevant during system failure or system changes. This study did not focus on sub-system failure rates and is a two-dimensional representation with a set speed so it does not capture detailed flight information, such as angle of attack or thrust settings, therefore the system monitoring meta-task would provide limited insight. Communication is just assumed to be out of scope. Communication is common taxiing, takeoff, and landing flight phases [60]. Since this scenario starts in the contested environment, communication is assumed to play a smaller role.

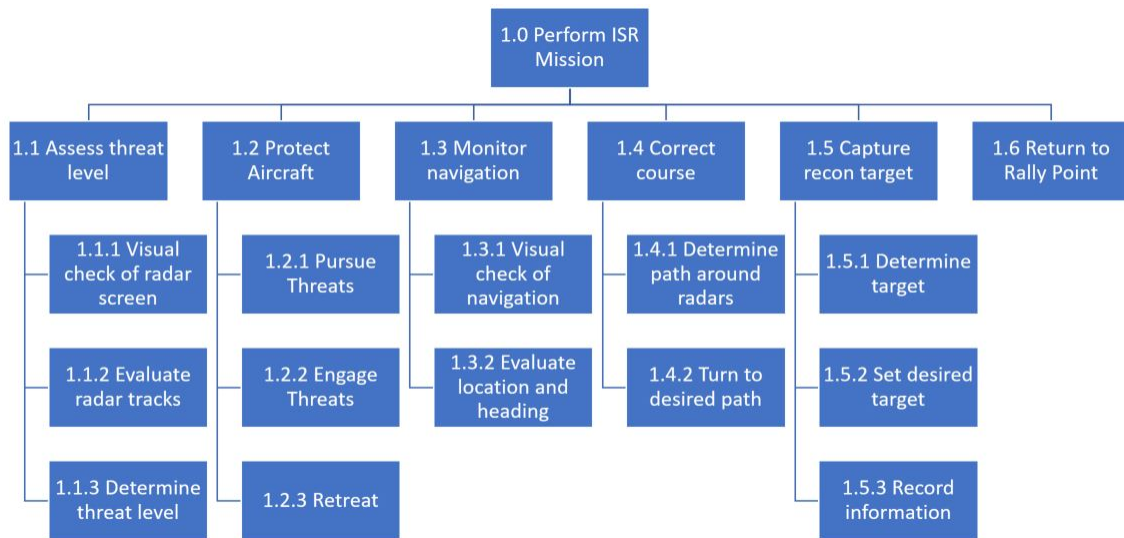


Figure 4.2: Basic hierarchical task analysis for performing an ISR mission in contested airspace

Although the HTA helps visualize the pilot's goals, it needs to be converted into a behavior tree for utilization in an agent-based model. Each goal and sub-goal creates a behavior in the dynamic task list, which the agent activates based on the awareness from the sensory model. The pilot's behavior tree is shown in Figure 4.3. Starting with the primary branch "assess threat", this behavior constitutes the operator checking the radar, acquiring threat information, and determining a threat level. The pilot cannot engage a threat until

they are made aware of its presence. Pilot's continually attempt to acquire information in the cockpit, so this behavior and the frequency of its activation can be driven by expected eye scan rates or distractions based on the sensory model. If the pilot determines a radar track poses a threat, then he activates the "protect aircraft" branch. The pilot is now focused on the threat and no longer focused on the ISR task. The pilot turns towards the closest threat. The pilot determines the intercept location based on where they expect the threat to be based on its heading and speed. Once within missile range, he launches a missile. The time that the missile is in the air, the aircraft and pilot are still considered threatened and the pilot monitors the missile until it hits or misses the desired target. The pilot stays in these two primary branches until the radar check determines that the blue aircraft is no longer threatened. Once it is not threatened, the pilot proceeds to the third primary branch of "navigate", where he acquires the aircraft's position and heading. The "aviate" branch uses the navigation information to find a route or turn to follow the route. If no route is currently selected, such as after an engagement or after a checkpoint is reached, then a route is found based on the current location and the locations of the red radars. The red radar locations must be avoided based on the expected detection distance. For the piloted aircraft, this distance is set to the 100 mile radius. The process of finding the route is assumed to be automated, however, the pilot must activate the action. If a route is already determined, then the pilot can activate the "turn to route" behavior if the plane's current location and heading is taking it off course. The "perform recon" branch controls the current ISR task. The pilot starts with checkpoint 1, and selects its location for "aviate" tasks. The pilot is able to "record information" as soon as "navigate confirms" the platform is in the proper location ("on station"). When all three recon checkpoints have been reached, the pilot turns to return to the rally point. When returning to the rally point, the pilot continues following the other behavior branches as appropriate.

The dynamic task list captures the basic actions taken by a pilot performing an ISR mission in contested airspace. It focuses on the pilot's behaviors as information is acquired

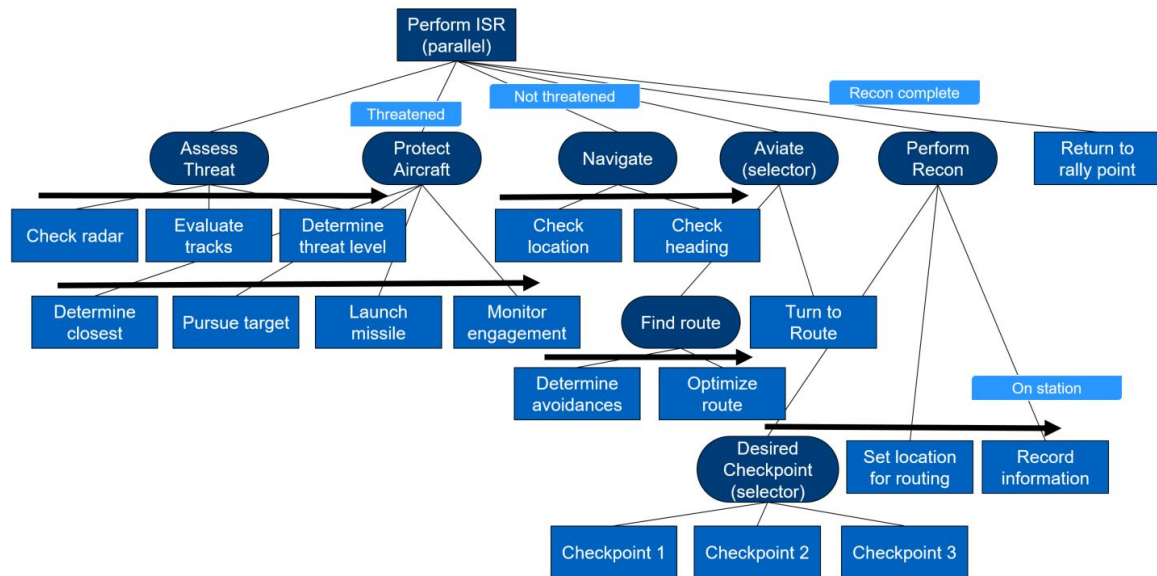


Figure 4.3: Representation of the behavior tree for the pilot agent created from the HTA breakdown

and goals may change throughout the operational environment. The dynamic task list also provides insights into how the operator is interacting with the system. The operator is performing specific actions to acquire and utilize information. The pilot must check the radar and navigation to increase his understanding. The complexity of the space can quickly expand the size of the dynamic task list. Further flight information could be included such as altimeter and fuel, as could actions related to the air-to-air engagement such as countermeasures and detailed maneuvering. However, this basic task list demonstrates how even a small piece of the operator workload helps decrease the uncertainty introduced by the inclusion of automation, but to capture the variation in workload, a few more pieces of the methodology must still be added.

4.1.3 Sensory Model

The sensory model captures the operator's understanding of his environment and controls the activation of the branches in the dynamic task list. However, compared to the highway domain, again the aerospace domain is more complex and less documented when assessing operator awareness. A pilot has many sensors focusing on awareness beyond the visual

range and is scanning a large volume with few focal points. Situational awareness for a fighter pilot is gained by cross-checking the on-board radar, radar warning receiver, flight members, support assets, visual acquisition, and navigation [33]. Since the operational model is a simplified one blue fighter scenario, the sensory model focuses on the on-board radar, visual acquisition, and navigation. The blue aircraft's radar detects red threats in the area, while navigation such as GPS specifies the blue aircraft's location and heading. The blue aircraft's radar is capable of detecting red aircraft out to 95 miles with a frame time of 0.5 seconds. It is a volumetric sensor with a probability of detection of 1.0. The radar's range is depicted in Figure 4.4. The radar purges track data if no information is acquired for 60 seconds. However, this represents the system's capabilities. The pilot only acquires information as required by the behavior tree. The behavior tree's update rate is set to one hertz. The parallel tasks at the top of the tree are evaluated every update. Any detection or track information from the radar is only utilized by the pilot when performing the threat assessment. This assessment occurs each time step because a typical step in the cross-checking scan by a pilot only takes a couple hundred milliseconds [120]. The pilot is also expected to perform visual scans outside the cockpit. Pilot's continually monitor the surrounding skies, but are only capable of detecting aircraft out to 10 nautical miles on a clear day and more likely only able to detect aircraft at a range of approximately 2 - 3 nautical miles [103]. A pilot's central vision is only 2 degrees so as distance increases the likelihood of detection decreases, unless the pilot is cued to the target (such as by the radar) so they are only searching a few degrees for a target [103]. The 10 nautical mile range is shown in 4.4. Since only missiles are modeled and the close air-to-air combat is not the focus, the visual component plays a small impact. The missile range is 50 miles and nothing limits launching at this distance, the visual detection does not have an impact on the track list. It is a workload task, but threat and missile tracking is done with the radar.

Based on the operations model and dynamic task list, navigation is the other situational awareness focus for the pilot that guides the agent behavior. Although the aircraft is al-

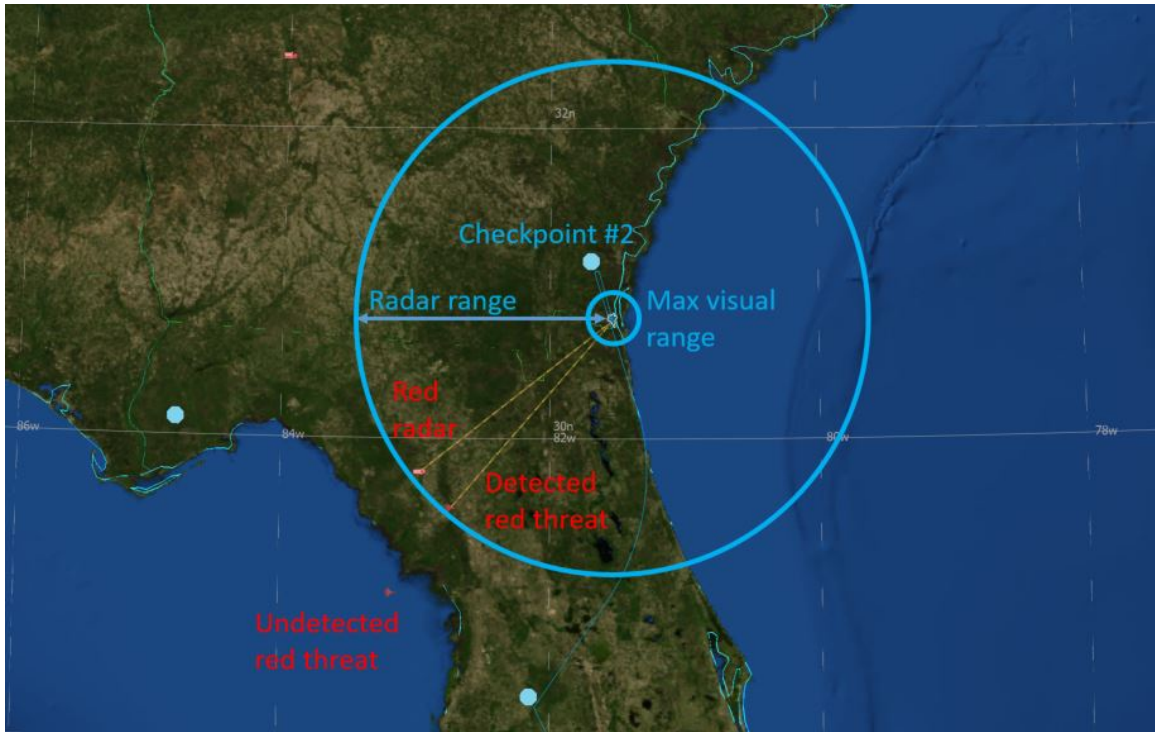


Figure 4.4: Blue aircraft sensory model showing the two-dimensional range for the radar and pilot's vision. The aircraft's current location allows one red fighter to be detected.

ways tracking threats and location, the pilot's attention is not always collecting all of the information. If navigation and radar are on separate consoles, then the pilot will rely on cross-checking to acquire all relevant information [120]. When the pilot is focused on a set of data, he is less likely to check the other sensor data. The F-16 Handbook discusses loss of awareness during prolonged engagements as the pilot focuses on the task at hand [33]. Therefore, the navigation task is not executed unless there is no perceived threat. The fighter freely travels through the space when engaging a target. When navigation is allowed, the pilot acquires his current latitude, longitude, and heading to be used in route finding. Additionally, if the pilot determines that his current location is within range of a radar, then he turns to egress detection as quick as possible before continuing towards the desired checkpoint. Based on the display cross-checking time of a couple hundred milliseconds, navigation, if allowed, is updated each behavior update [120]. The radar and navigation aspects capture the pilot's awareness instead of the entire aircraft's, which drives

the execution of the dynamic task list and blue behaviors inside the operations model. The last piece of the methodology is translating the operator's task list and sensory elements to a dynamic workload.

4.1.4 Workload Model

The workload model creates a dynamic measurement of the operator's workload. The three focus categories are the mental demand, temporal demand, and performance. Mental demand represents the mental activity required throughout the operation, while temporal demand relates to the time pressure or pace of the tasks. The number of cars in memory created a good stand in for the driver's mental workload, because it represented the number of elements that the driver was paying attention to. However, for a pilot, they are continually searching for targets in a vast space, with minimal detections, therefore number of detections alone does not present a good metric. The effort to acquire and maintain situational awareness has shown to have a correlation to workload [65]. Additionally pilot awareness has shown to decrease with increased maneuvering and during an engagement [33]. Combining these details, the mental workload can be partially quantified using the maneuvering data. Since the model is only two-dimensional, this is quantified using required turn data. A high-rate turn represents a high mental workload as the pilot ascertains information about his new heading and relative bearing to threats. While straight travel results in lower mental demand, because the cross-checking is looking for undesirable changes in awareness rather than developing awareness. The temporal demand then can be calculated based on the frequency of turns, which represents the rate at which the pilot is acquiring new information. Frequency is determined based on the time spent maintaining the same heading. A high frequency correlates to a high temporal demand, therefore, temporal demand is inversely related to the time between turns. Lastly, performance can be calculated based on the time the aircraft takes to return to the rally point compared to the optimal time to perform the ISR mission. The optimal time is calculated based on the aircraft's speed and the path of

minimum distance through the checkpoints. The optimal path is 815 nautical miles long, which is the straight-line distance between the rally point to Checkpoint 1 to Checkpoint 2 to Checkpoint 3 then back to the rally point, and the aircraft's speed is 600 knots. Therefore, the time used in performance calculations is 4,890 seconds. Although the current blue aircraft is unable to fly this path, this represents the time required for undetectable, invincible aircraft. To convert the performance to a percentage, the flown time is divided by the ideal time with the difference from 100% representing the performance loss rate (ex: if the division produces 160% then the performance loss rate was 60% resulting in a performance of 40%). This method of evaluating performance ensures a common divisor. The mental demand, temporal demand, and performance represent the first set of insights provided by the workload model, however, the detailed action list and sensory model enable further operator-focused analysis.

The track and task frequency data are the second component of the workload model. They provide insights to the pilots actions throughout the mission. The focus is on the time spent by the pilot in the engagement, tracking both opposing aircraft and launched missiles. The time spent threatened demonstrates a discontinuity in the operator's awareness [33]. The dynamic task list does not allow navigation during engagements. The quantification of this time and the number of tracks enables the decision maker to assess level of awareness and any impacts the awareness and current action set may have on operational effectiveness.

4.1.5 Technology Integration

Experiment 4 shows how the pilot's workload changes when a UCAV is added. The four components described above capture the piloted aircraft performing the ISR mission, however, for the UCAV to be assessed by the methodology minor changes to the model must be made. The addition of automation requires assignment specific tasking between the automation and the operator, along with modifications to the operator's dynamic task list to capture the monitoring and correction behaviors. Looking first at the tasking, the UCAV is

currently intended to be paired with a piloted aircraft that assigns tasks and monitors functionality [79]. The UCAV will likely need to be largely automated with a management-by-consent control strategy [79]. In this control strategy, the UCAV determines a set of desired actions then the pilot must approve them before executing the decision. Additionally, a primary benefit of the UCAV system in contested airspace is its proposed low radar signature [41]. A lower radar signature reduces the probability of detection and can open new pathways through a radar network. The radar range equation captures the expected performance of a radar. Simplifying this equation down provides the relationship between radar cross section and the maximum detection range

$$R \propto RCS^{-(1/4)} \quad (4.1)$$

where R is the radar's range and RCS is the detected aircraft's radar cross section [103]. Assuming that the UCAV has an RCS that is half as large as the manned-aircraft, this results in a radar detection range approximately 88% of the magnitude specified in the operations model. The effect of the decrease in radar range is shown in Figure 4.5. There is now a more direct pathway between Checkpoint 2 and Checkpoint 3 that the UCAV can use but the piloted aircraft cannot. This difference in capability and to maximize safety and efficiency means the pilot should task the UCAV to the ISR checkpoints while he monitors from a distance. However, since the team needs to communicate, the pilot must maintain line-of-sight. Although there are alternative communication methods, in contested airspace it is assumed a technology similar to Link 16 would be used, which is currently limited to line-of-sight [104]. Therefore, it is assumed that the piloted aircraft flies a combat air patrol (CAP) within LOS but not through the checkpoints while the UCAV flies the ISR mission. According to MIL-STD-3013, a CAP occurs at an altitude of 35,000 feet, while a low-level ISR mission occurs at 2,000 feet of pressure altitude [76]. Line-of-sight (LOS) is calculated assuming a spherical earth using the following equation:

$$LOS = 1.06 * (\sqrt{h_1 * (2 * R + h_1)} + \sqrt{h_2 * (2 * R + h_2)}) \quad (4.2)$$

where h_1 and h_2 represent the altitudes of the two aircraft, 1.06 captures the 6% increase in range due to atmospheric refraction, and R represents the Earth's radius (assumed to be 6,378 kilometers) [84]. This provides a maximum LOS distance between the UCAV and the piloted aircraft of 261.7 nautical miles. These modifications now must be included in the dynamic task list to control the team's behaviors.

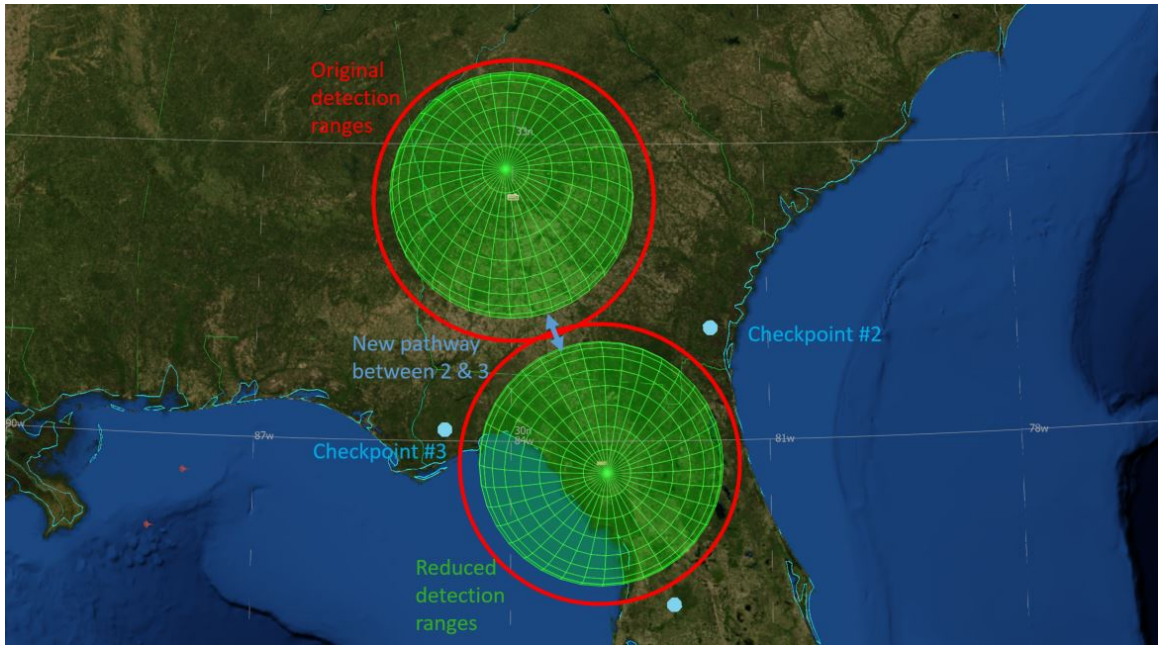


Figure 4.5: A comparison of radar detection ranges. The original range is for the piloted aircraft, while the smaller range is used for the UCAV due to its assumed smaller RCS.

The dynamic task list must be modified because although the operator's overall goal remains the same, some sub-goals move to the UCAV while additional sub-goals are added to control the UCAV which effects the behavior tree. The updated behavior tree is shown in Figure 4.6. The yellow boxes highlight changes in the pilot's behavior tree. The reconnaissance task of moving between checkpoints is now handled by the UCAV, but the pilot must monitor the UCAV as it navigates the contested airspace and correct its behavior (approve assignment to the next task) when it reaches a checkpoint. The monitoring task

attains the UCAV's current position and heading. When this check occurs, the pilot evaluates the UCAV's distance from the desired checkpoint, if it is at the desired checkpoint then the pilot approves it continuing on to the next checkpoint. The activation frequency of the perform recon behavior is determined based on the pilot's trust and understanding of the automation. A low-trust or experienced operator more frequently checks on the automation [56]. The baseline checking frequency assumes that the UCAV becomes part of the pilot's cross check and therefore it runs every update cycle unless the pilot feels threatened. Based on where the UCAV falls in the sub-goals and resulting behavior tree, the piloted aircraft ensures the safety of his aircraft prior to checking on the UCAV. The addition of the automation also changed the piloted aircraft's navigate and aviate tasks. Navigate now ensures the distance between the UCAV position (based on the last monitor time) does not exceed the maximum distance, while aviate has changed its route planning to create a suitable CAP for monitoring the UCAV.

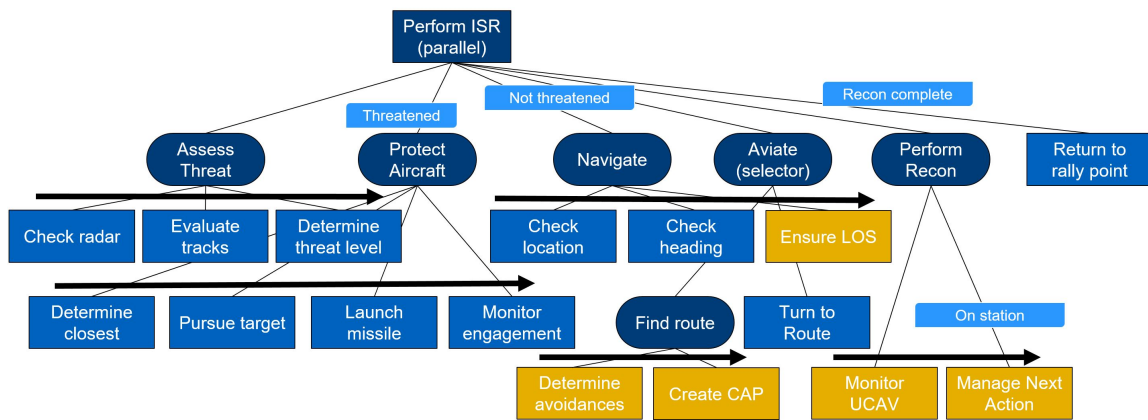


Figure 4.6: Modified behavior tree for a piloted aircraft with a taskable UCAV

Lastly, the sensory model must include the new actions in the dynamic task list. The time required by the pilot to acquire the new pieces of information and how long he holds the information must be defined. Both the monitor and manage steps require operator awareness. The monitor step is assumed to be completed in a couple hundred milliseconds similar to radar and navigation checks. The management task is assumed to take longer and likely be variant because the pilot must evaluate the UCAV's current state and approve

its next action within a environment of varying complexity. Since information could not be found on the expected complexity of this interaction, a lognormal distribution with a mean of 1.2 seconds and a standard deviation of 0.36 seconds is utilized based on the mirror check data from driving [62]. All UCAV information is assumed to be checked and stored only briefly in the pilot's memory until the next behavior update decides to check it again. The effects of the technology inclusion is shown in Figure 4.7. The thin blue lines show the UCAV and manned-aircraft route. The UCAV is now routing around the smaller radar ranges performing the ISR component. The pilot sets up a CAP over approximately Checkpoint 1 where he remains within LOS of the UCAV. When the pilot is threatened, he engages the red aircraft then returns to his CAP. With the model defined both with and without the UCAV, workload results can now be collected and compared.

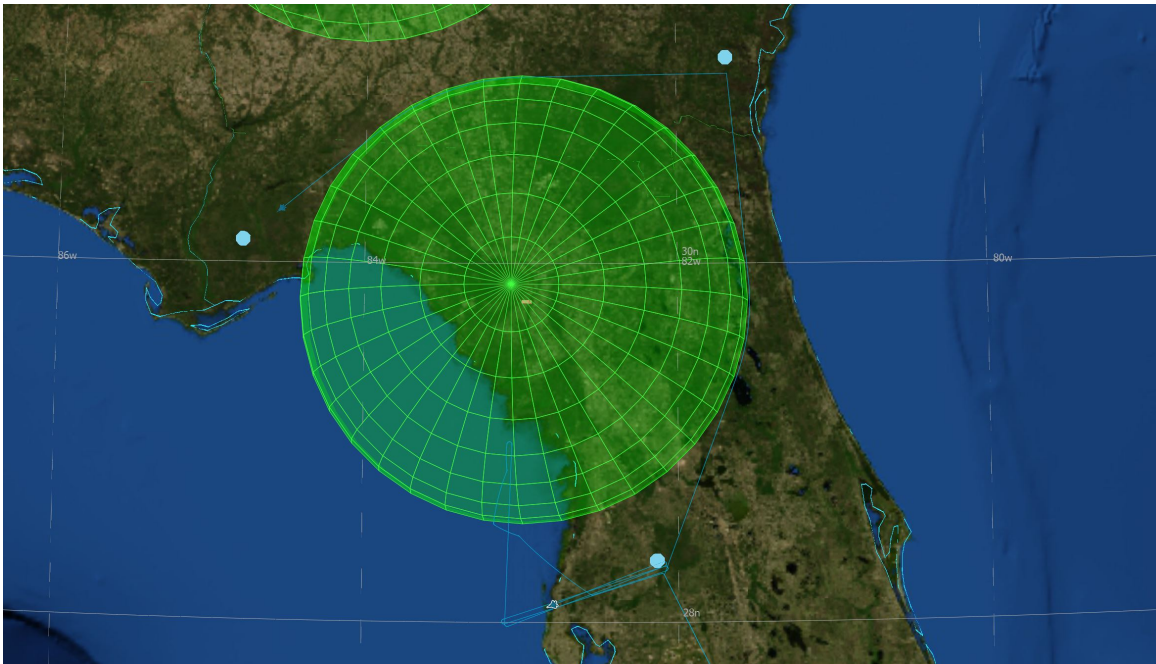


Figure 4.7: Visualization of operational environment after the inclusion of automation technology (UCAV). The thin blue lines show the blue aircraft routes throughout the mission.

4.2 Experiment 4 Results

This experiment tests the methodology's capability to assess manned-unmanned teaming while performing an ISR task. Figure 4.8 shows the an action summary for the pilot performing the ISR task independently. The lines at the bottom of the graph capture the pilots actions throughout the mission. The dashed line shows the time that the aircraft arrives at each checkpoint: Checkpoint 1 is reached at 911 seconds, Checkpoint 2 is reached at 1,989 seconds, and Checkpoint 3 is reached at 4,392 seconds. The long duration between Checkpoints 2 and 3 is due to the air-to-air engagement. This engagement (shown as the orange line) occurs from 2,117 seconds until 2,686 seconds. The pilot is engaged by one aircraft, but acquires a second track as he turns to engage (the yellow line). The blue peaks show the duration of time in flight for each missile fired. This action breakdown becomes important when the technology is introduced, because the pilot is assumed unable to monitor the UCAV while it is threatened. The top blue line on Figure 4.8 shows the instantaneous heading of the aircraft throughout the mission. Shown are the navigational changes conducted by the pilot while avoiding red radar sites and engaging red aircraft. This heading data is utilized to understand the pilot's workload throughout the mission.

The workload model section discussed how the three workload metrics (mental demand, temporal demand, and performance) are calculated. Starting with mental demand, it is assumed equal to the magnitude of the turn rate due to the pilot's workload in gaining a new state of awareness as he turns. Figure 4.9 shows the absolute value of the turn rate throughout the ISR mission. The maximum turn rate is 5.92 degrees per second occurring after the pilot reaches Checkpoint 1 and turns sharply towards Checkpoint 2 while avoiding the radar site. The time-based average mental demand is 1.03 degrees per second and shown on the dashed line on Figure 4.9. The temporal demand has an inverse relationship with the variations in the turn rate. A variation is calculated as the time between a change in the turn rate. The average time between variations is 8.79 seconds, which correlates to

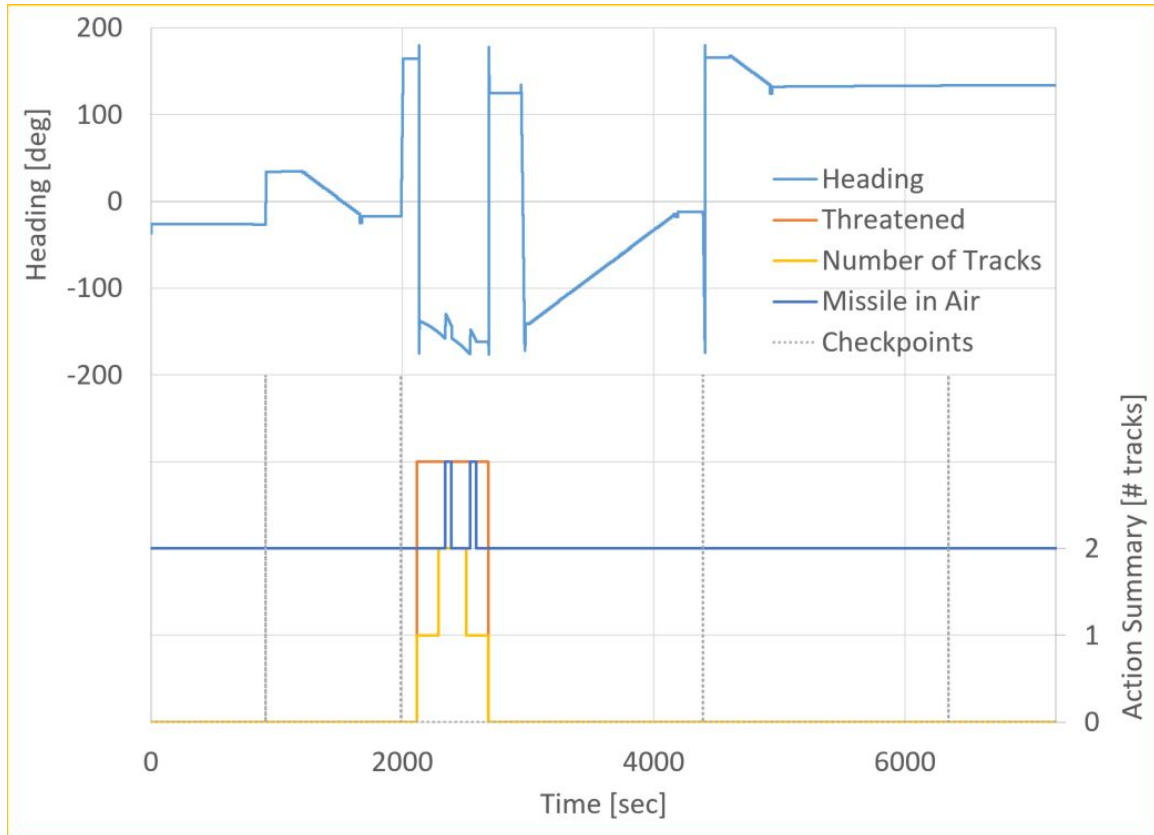


Figure 4.8: Summary of pilot actions while performing ISR mission solo.

a temporal demand of 0.11 sec^{-1} . Lastly, the time for the pilot to complete the ISR mission is 6,347 seconds. Comparing this value with the ideal time of 4,890 seconds results in a performance of 70.2%. These three workload metrics provide benchmark values when assessing automation technologies.

Currently, only one technology is being assessed, a UCAV conducting manned-unmanned teaming with the pilot. The UCAV conducts the surveillance tasks, independently avoiding radar sites and navigating to the checkpoints. The UCAV is assumed to have a radar cross-section half the size of the piloted aircraft so it is able to choose a different route than the piloted aircraft did. The pilot maintains a loiter position within line-of-sight of the UCAV to monitor its progress, and engages the red aircraft. After reaching each checkpoint, the UCAV loiters until the pilot monitors the data collection and approves it moving to the next destination. A summary of the pilot actions with the UCAV teammate is shown

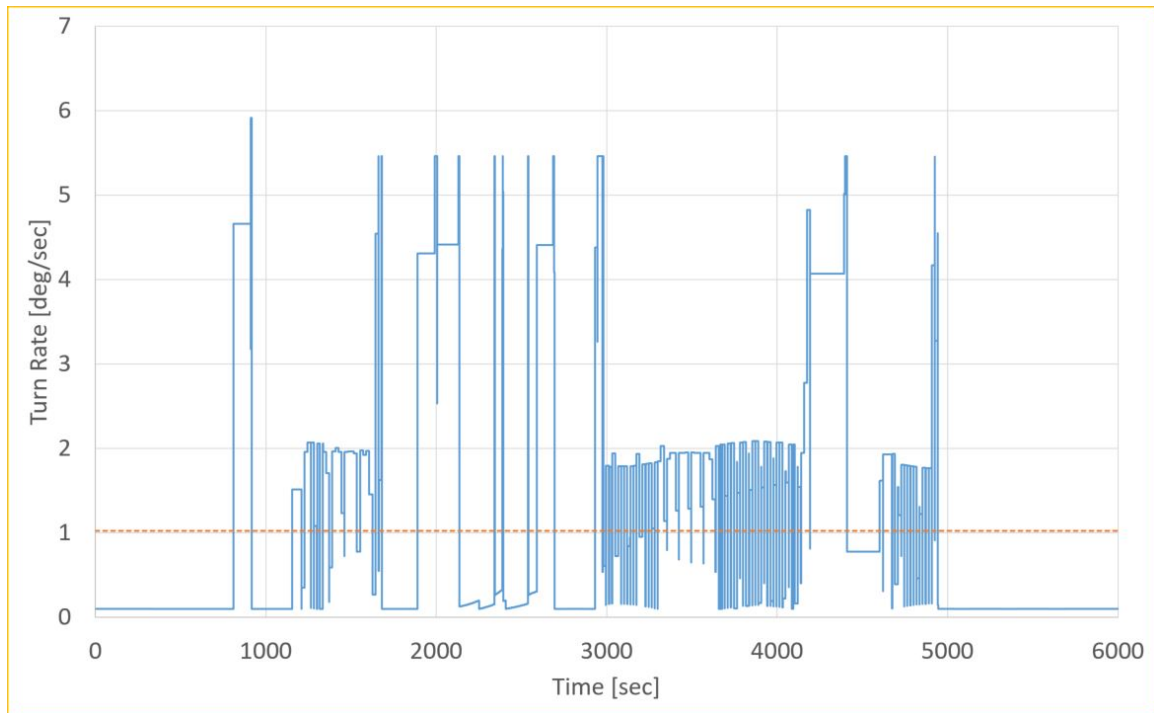


Figure 4.9: Pilot turn rates while performing the ISR mission solo. The average magnitude is shown as the dashed line at 1.03 degrees per second.

in Figure 4.10. The same action pieces have been included as in Figure 4.8 to capture when the pilot is distracted by the engagement. The dashed line again shows the time that the aircraft arrives at each checkpoint: Checkpoint 1 is reached at 911 seconds, Checkpoint 2 is reached at 1,931 seconds, and Checkpoint 3 is reached at 3,192 seconds. The UCAV reaches Checkpoint 2 quicker due to the shorter route around the red radar, and the time between Checkpoints 2 and 3 is reduced because of the new route between the radars enabled by the reduced RCS. Additionally, the loiter time for the UCAV is shown, representing when it is awaiting approval by the pilot. From 1,931 seconds to 2,168 seconds, the UCAV is awaiting the pilot to finish the engagement. The pilot's heading is more step-wise than the solo pilots navigation, because he is maintaining a CAP within line-of-sight of the UCAV rather than flying the route himself. These changes result in variations to the workload metrics.

The mental demand for the pilot utilizing a UCAV is shown in Figure 4.11, again represented by the maximum value of the turn rate. The average mental demand (shown by the

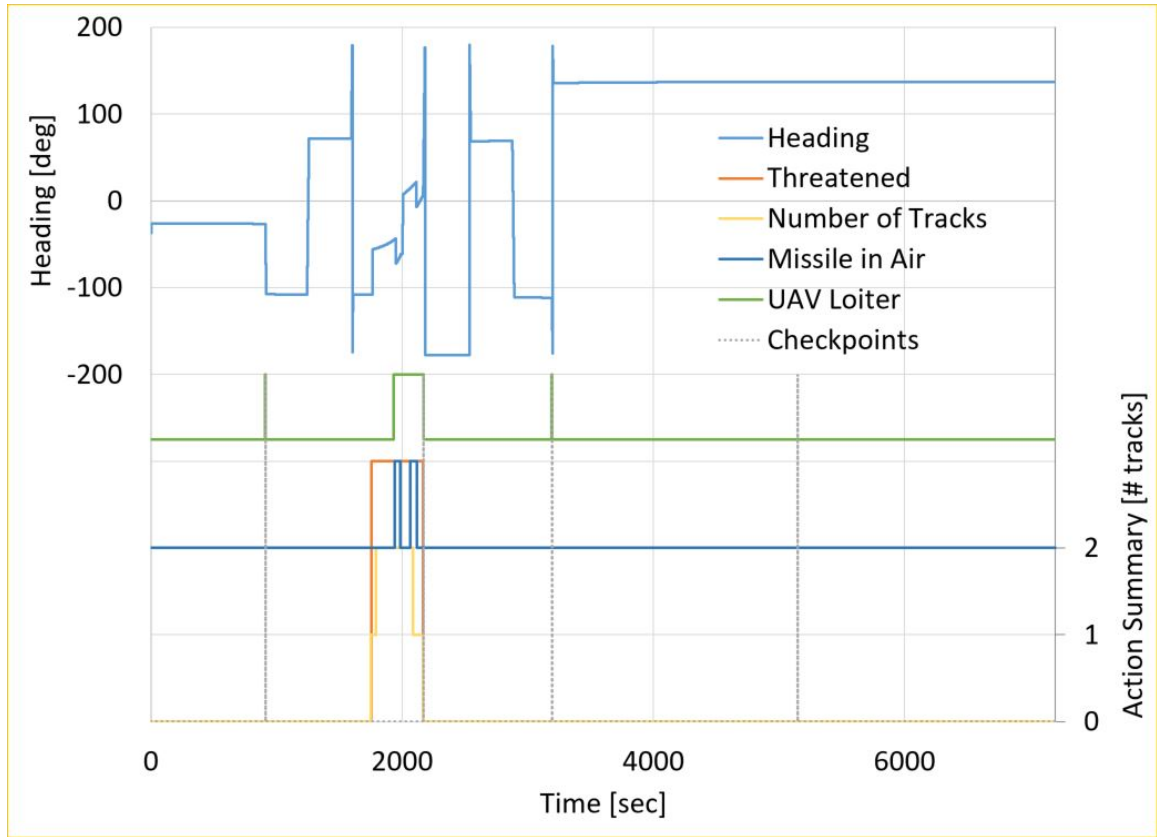


Figure 4.10: Summary of pilot actions while performing ISR mission jointly with UCAV (RCS is half of piloted aircraft).

dashed line) has decreased from 1.03 to 0.56 degrees per second. However, the maximum turn rate of 5.92 degrees per second (the limit based on the set speed and g-limit of the piloted aircraft) is more frequent in occurrence and the duration is longer. The multiple peaks is due to the 180 degree turn conducted at the edge of the CAP requiring high mental workload to regain awareness followed by a lull while the pilot continues in a straight path, instead of the continuous navigation around the contested airspace. This flight pattern also results in an approximately 10% decrease in the average temporal load of 0.11 sec^{-1} for the solo mission (a mean time between variations of 8.79 seconds) compared with 0.10 sec^{-1} for the mission with the UCAV (a mean time between variations of 9.62 seconds). Lastly, the UCAV had a substantial impact on the mission performance. The solo mission was conducted in 6,347 seconds, while the inclusion of the UCAV reduced the mission time to 5,148 seconds. Comparing this with the ideal mission time of 4,890 seconds results in a

performance of 94.7%. The 237 seconds that the UCAV spends waiting for pilot approval is the primary performance degrade, because the checkpoint layout allows the UCAV to fly a near optimal route with its decreased RCS. The decrease in mental workload and temporal demand while improving performance is the expected behavior for the team because the tasks are split between the UCAV and the pilot. The UCAV enables one pilot to increase his performance, thus decreasing the military personnel requirement. The loss in performance waiting for pilot approval is negated by the improvements from a second aircraft with a reduced RCS.

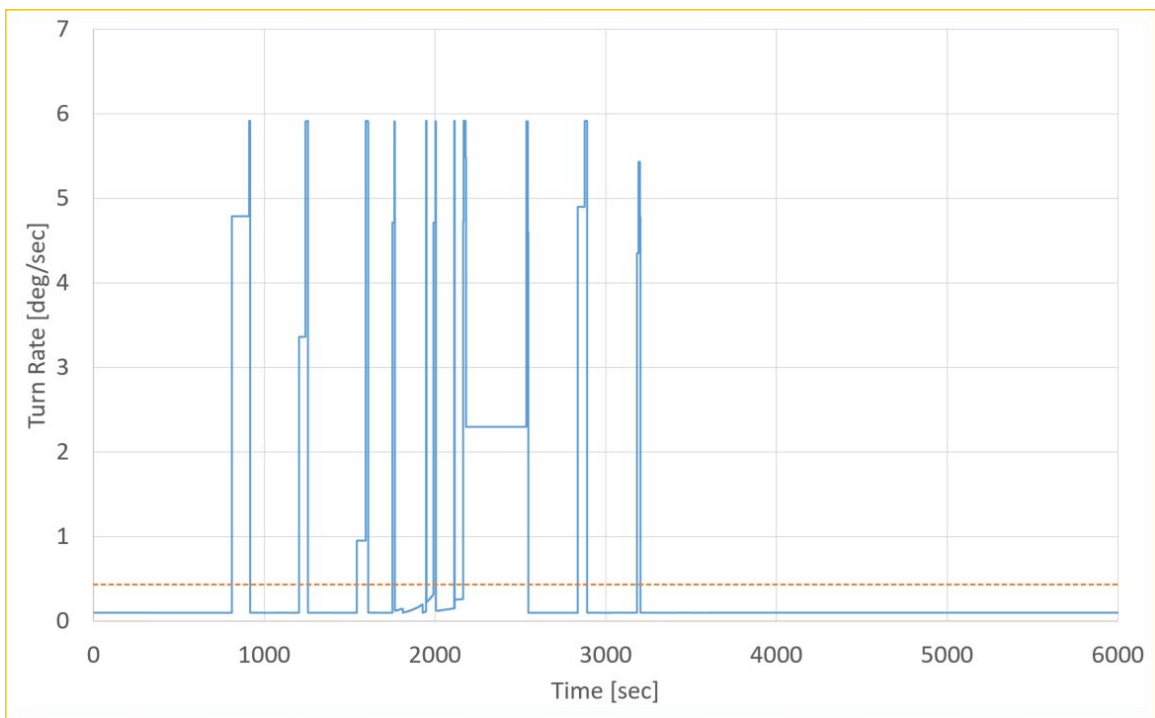


Figure 4.11: Pilot turn rates while performing the ISR mission with the UCAV (RCS is half of piloted aircraft). The average magnitude is shown as the dashed line at 0.56 degrees per second.

However, since the UCAV is still in development, the system design is not complete. Therefore, it is important to look at the technologies design along with pilot workload. The RCS of the UCAV has been assumed to be half of the piloted aircraft, however, Figure 4.12 captures the action list for the pilot if the UCAV maintains the same RCS as the piloted aircraft. The same air-to-air engagement scenario exists for the piloted aircraft, but the

mission now takes longer as the UCAV must take the same path around the radar sites as the piloted aircraft. The UCAV's checkpoint arrival times are now: Checkpoint 1 is reached at 911 seconds, Checkpoint 2 is reached at 1,987 seconds, and Checkpoint 3 is reached at 4,387 seconds. Checkpoints 1 and 2 are reached in similar times as the piloted aircraft (the small delay for Checkpoint 2 comes from the approval time required after arrival at Checkpoint 1). The UCAV then waits at Checkpoint 2 between 1,987 seconds and 2,168 seconds before the pilot finishes the engagement and can approve the UCAV's next action. However, where this waiting time is made-up in the faster routing for the UCAV with a reduced RCS, it is lost performance for this scenario, which can be seen when looking at the workload metrics.

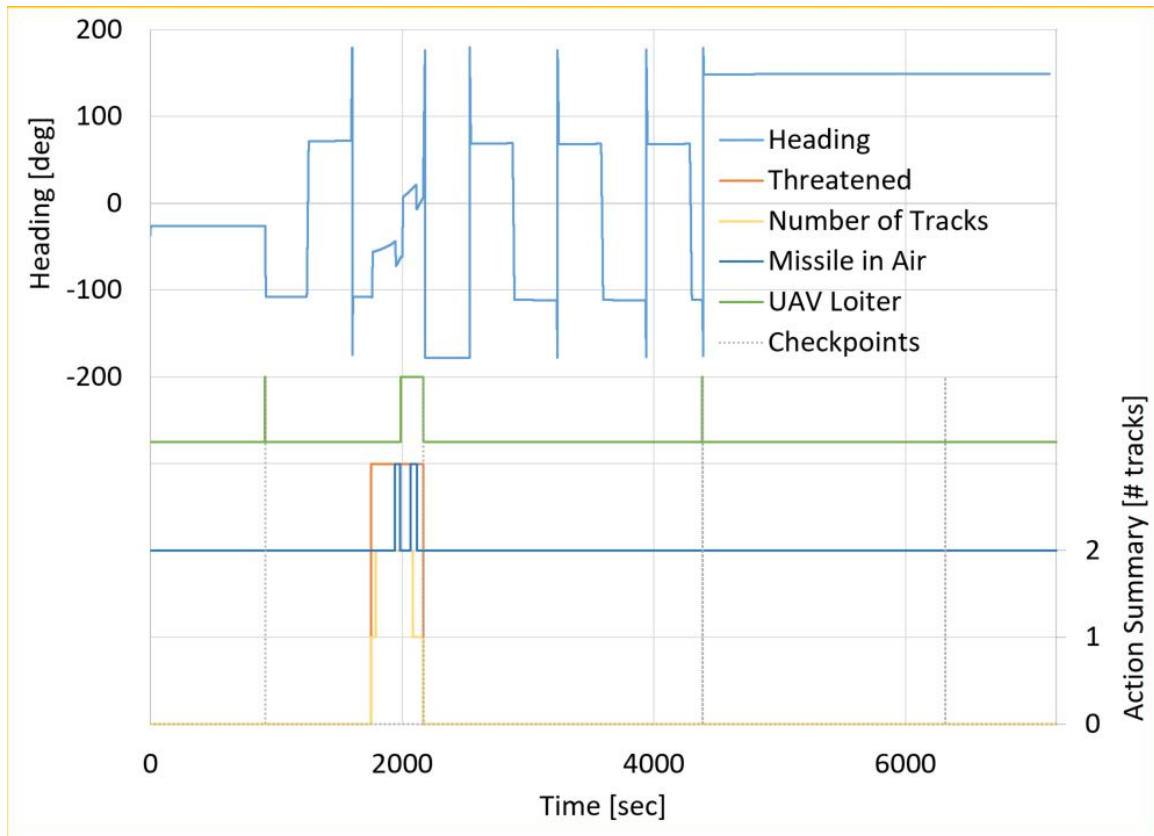


Figure 4.12: Summary of pilot actions while performing ISR mission jointly with UCAV (RCS is equal to the piloted aircraft).

Changing the UCAV's RCS impacts the three workload metrics. Figure 4.13 shows the mental demand for the pilot utilizing the UCAV, captured by the maximum value of the

turn rate. The average mental demand (shown by the dashed line) remains almost the same for both UCAV cases, slightly increasing from 0.56 to 0.58 degrees per second. Similar behavior is also seen in the frequency and duration of the 5.92 degrees per second turn rate, with the pilot utilizing the UCAV requiring quick turn-arounds to maintain his CAP. The change in the UCAV results in an approximately 8% decrease in the average temporal load of 0.104 sec^{-1} for the mission utilizing a UCAV with half the RCS (a mean time between variations of 9.62 seconds) compared with 0.096 sec^{-1} for the mission with the UCAV at the same RCS (a mean time between variations of 10.4 seconds). The average temporal demand decreases because the pilot spends more time maintaining his CAP, which creates a relatively low temporal load. This increased time in the CAP correlates to a lower overall mission performance. This team completes the mission in 6,315 seconds, which is only slightly faster than the 6,347 seconds taken to complete the solo mission. This results in a performance of 70.9%. The piloted aircraft is able to return to managing the UCAV as soon as he is not threatened. Therefore, while the pilot would be focused on egressing the radar coverage before proceeding in the solo mission, the UCAV is able to continue towards Checkpoint 3 while the pilot egresses towards his CAP. These workload metrics enable new insights for assessing manned-unmanned teams and can be utilized to highlight areas for increased analysis.

4.2.1 Trade Space

The data collected from the mission with and without technology provides the basics of a tradespace, and demonstrates the benefits of the new methodology when evaluating automated technologies in the aerospace domain. Figure 4.13 compares the workload metrics captured above. They have been normalized by the solo mission data to show comparative trends. The inclusion of the UCAV created an almost 50% reduction in the operator's mental demand due to navigation regardless of the UCAV's radar signature. Similarly, the temporal demand (pace) of the navigation tasks reduced between 9% and 16%. However,

performance is variant depending on the UCAV's signature. A UCAV with the same signature of the piloted aircraft has almost the same performance, while a UCAV with half the RCS increases the performance by 35%. Figure 4.14 compares these values in a trade-off environment. Although this environment is only the start of an entire tradespace, it demonstrates the pros and possible cons of including automation technology. In this scenario, the inclusion of the UCAV reduces the operator's workload and has the potential of increasing performance. However, the performance increase is done at the cost of adding an additional aircraft driving up the mission's maintenance and operating costs (not modeled explicitly). This may be a trade that the decision maker is willing to make, however, the oval for the UCAV's performance captures the variability due to RCS. The increased cost of two aircraft may not make sense if the RCS remains the same as the piloted aircraft. Alternatively, the mental and temporal workloads due to navigation decreased when the UCAV is utilized therefore the pilot may switch to a front line operator. The visibility into the operator's task list and awareness enables better comprehension of the effects of teaming.

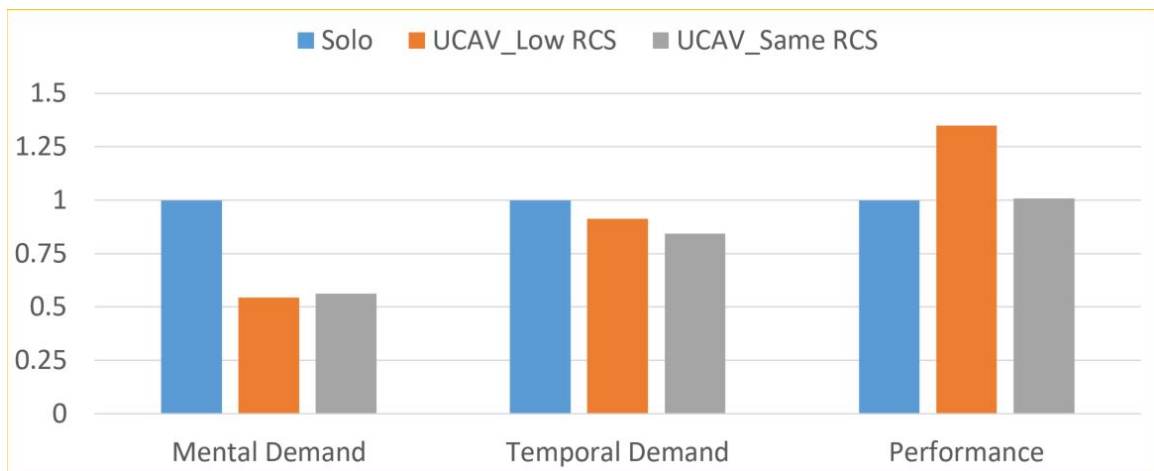


Figure 4.13: Normalized comparison of average workload metrics (normalized by the results of the solo case).

Although the task of monitoring the UCAV is now explicit in the model, it is not captured in the workload because it is assumed to be equal to the current instrument scan conducted by the pilot without the automation technology. However, the monitoring and

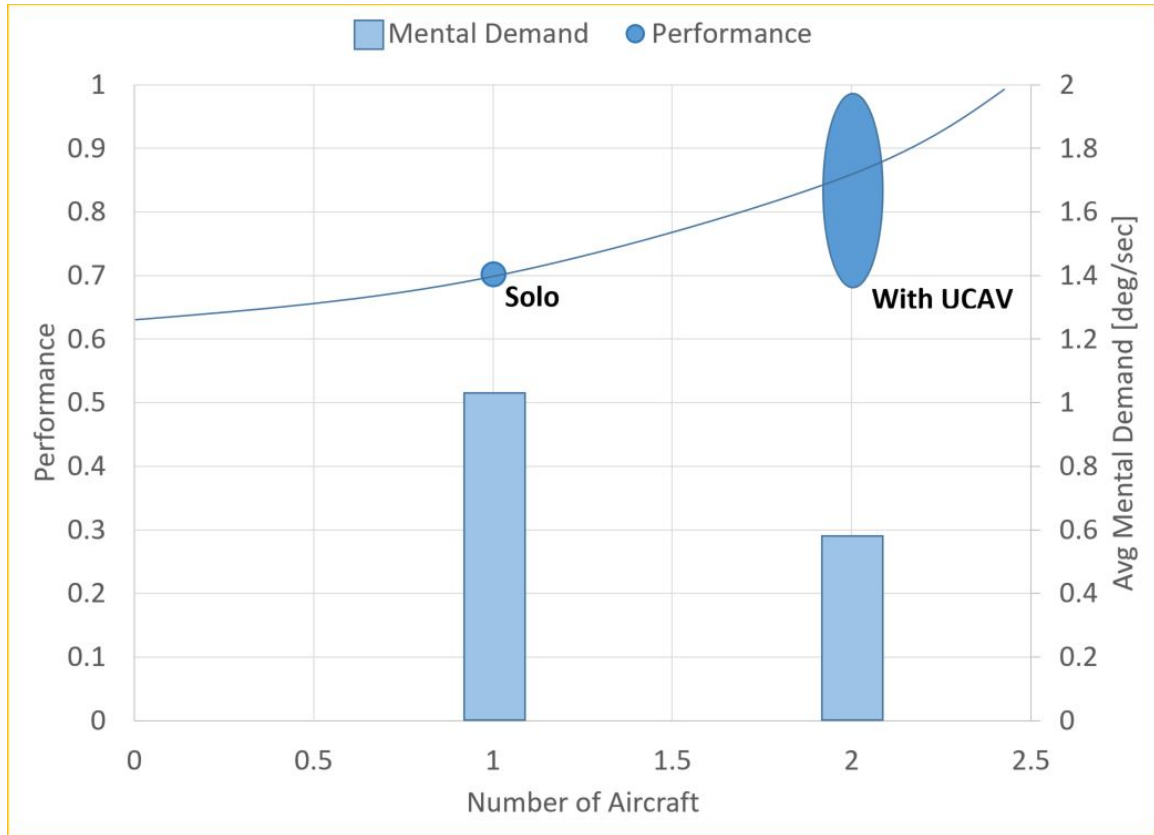


Figure 4.14: Normalized comparison of average workload metrics (normalized by the results of the solo case).

management branches are only conducted by the operator when he is not threatened. This explicit modeling of the operator enables a greater understanding of downtime created between team members. While the pilot is engaged, the UCAV waits approximately three minutes loitering over Checkpoint 2. The decision maker must decide how these partially automated system's behave when the manager is busy or unable to intervene in time. Similar to the traffic model, the pilot trust can effect the awareness of the pilot. Currently, the pilot is assumed to check the UCAV with each cross-check, however, over time the pilot may become more reliant on the UCAV's decisions. Since this operational context only has three checkpoints, trust was found to play a minor impact on the workload metrics with a somewhat trivial answer being directly correlated to the number of checkpoints. To best capture trust, there must be a continual desired action that the operator is monitoring. An example for this scenario is ensuring that the UCAV continues on an optimal path around

the radar sites, else the pilot should intervene and redirect the UCAV based on the pilot's awareness. This expansion of the scenario was deemed to require too many assumptions about the control methods and failure modes for the UCAV so is currently not modeled.

Overall, Figure 4.13 and Figure 4.14 demonstrate the capabilities of the new methodology to provide operator-focused insights in the aerospace domain. This experiment demonstrates the promise of the methodology by creating agents that are more representative of the fielded system. The behaviors are guided by cognitive studies and the operator's awareness is neither omni-present nor instantaneous. The result is a better understanding of the operator's workload and possible challenges when the manager is unavailable or unaware. This resulted in the inclusion of the UCAV decreasing the operator's workload while either maintaining similar performance or increasing performance. Additionally, the behaviors are now explicit between team members (the UCAV and the pilot) enabling an understanding of when and how they are communicating versus need to communicate. Therefore, Hypothesis 4.0, stating that the operator-focused methodology is capable of decreasing uncertainty in the analysis of manned-unmanned teaming, is accepted.

CHAPTER 5

DISCUSSION AND CONCLUSION

This work demonstrates a new methodology for conducting system analysis on partially automated systems. The inclusion of automated systems creates an uncertainty in operator workload and tasking. When automation is included, there is an implicit teaming between the operator and the technology which is currently not captured in early systems analysis. Automation is capable of out-pacing an operator or requiring little interaction from the operator, both creating possible awareness and workload concerns. Therefore, this work pairs cognitive engineering concepts with operations research to create an operator-focused agent-based model enabling systems assessments utilizing automation technologies. The methodology creates dynamic, traceable operator workload metric based on the criteria for a good system requirement. The components of the methodology are summarized in Figure 5.1. This methodology starts with an operations model, which is created using the 7-step OR process. This process helps scope the problem to a desired environment to assess the new technology, thus defining the necessary elements. Technology rarely changes the operational context of a system, however, when capturing operator and technology awareness the background elements must be captured, which may previously have been ignored. The build-up approach used by agent-based modeling enables the analysis of emerging, complex system behaviors. However, the internal and external components of an agent's taxonomy are often defined based on system capabilities, thus the actions of the operator versus the technology become ambiguous and workload unmeasurable. To overcome these issues the focus agent (the agent designed to capture the operator utilizing automation technology) must be developed using concepts from cognitive engineering. Therefore, this component defines and creates the environment elements of interest, but the focus agent is further broken down in the next two components of the methodology. These elements

include: the performance and behaviors of background agents (any agent that is part of the desired layout, but not the focus of the workload analysis), and the focus agent's system-level capabilities (performance constraints, sensors, weapons, etc.).

The next two components of the methodology add definition to the focus agent based on concepts from cognitive engineering. First is the dynamic task list maps the system capabilities to actions through operator's intent and awareness. This is done using the goal and sub-goal breakdown conducted in a hierarchical task analysis to create an operator-focused behavior tree. An operator event sequence diagram can be utilized to map goals to awareness elements to action tasks, which is a similar format to the coding of a behavior tree. When technology is included, the action set is shared between the automation and the operator. The HTA high-level goal(s) should remain the same, however, the operator's sub-goals may change including sub-goals related to managing the automation technology. This change in sub-goals drives the behavior tree changes. It moves some actions to a behavior tree for the technology and adds branches to the operator's behavior tree for new expected behaviors due to technology management or changing sub-goals. The second piece of agent definition comes from the sensory model. The sensory model breaks the tie between system capabilities and operator awareness. The operator must seek information and there is a time required to gain awareness. Seeking information is controlled by sensory model's interaction with the dynamic task list. The tie between awareness and behaviors ensures operator's are not omnipresent. Unlike a sensor that continually scans a space, an operator's behaviors should control when he is collecting information. The frequency of checks, duration of checks, and what data is held in memory by the operator should be modeled separately from that of system capabilities. Variations in these parameters helps capture system behavior changes due to operator trust. As trust increases, the operator will be less aware of the automation actions and decrease his ability to intervene. This correlates to a reduction in checks and can change the type and duration of automation related information in the operator's memory. The details of the sensory model are guided

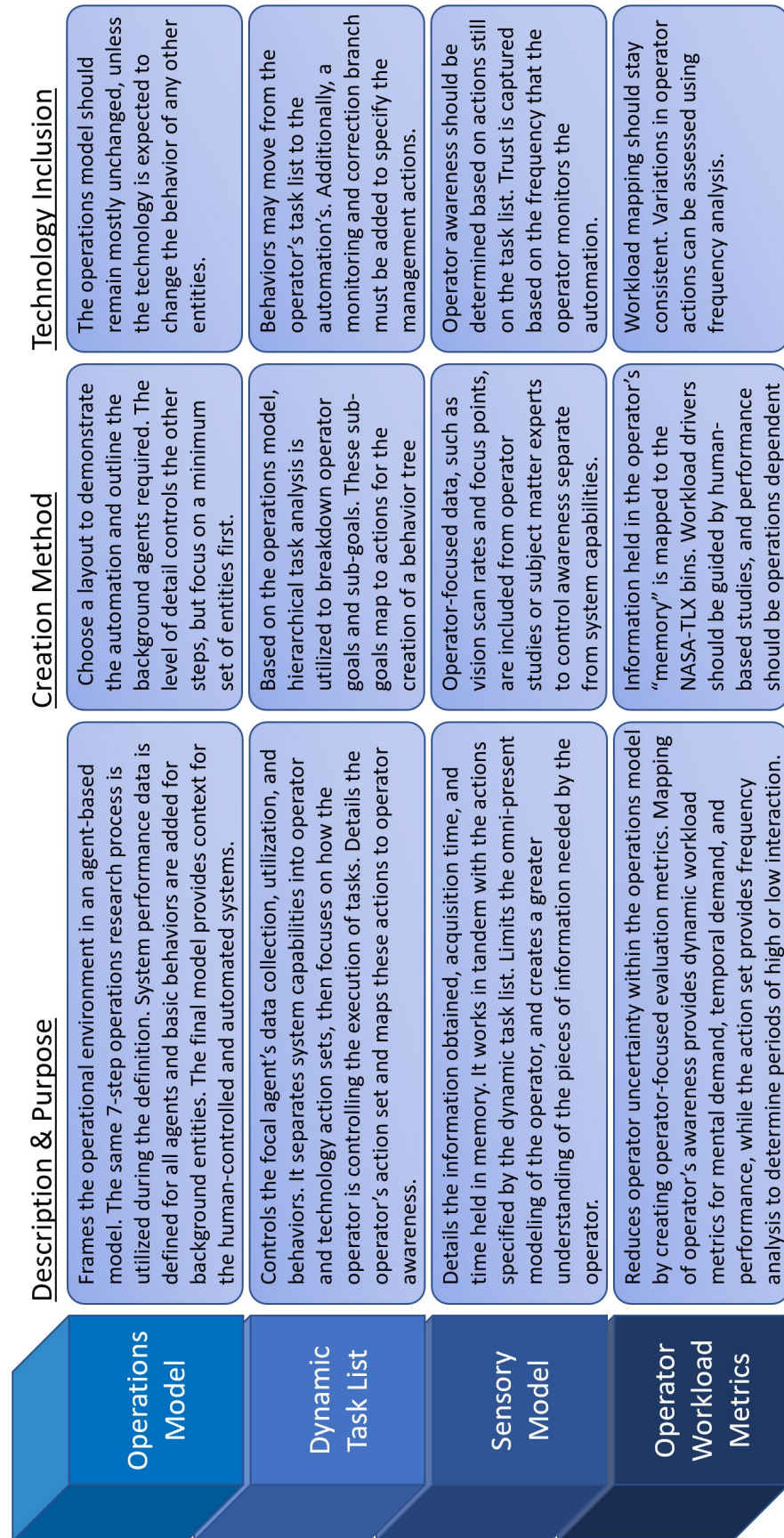


Figure 5.1: Overview of model components in methodology

by physical experiments regarding situation awareness. The creation of the sensory model is essential for capturing the operator as a limiting element of the system's success, and a key enabler for understanding the impact of introducing automation technologies early in the design process. Together the dynamic task list and sensory model separates the operator from the system to enable a dynamic workload assessment.

The final component in the methodology shown in Figure 5.1 is the workload measurement metrics. The current methodology focuses on three workload categories: mental demand, temporal demand, and performance. Mental demand captures the current working memory being utilized by the operator, while temporal demand captures the amount of workload felt due to the pace of the activities. The workload factors are unique to each case study, and should be guided by key drivers found in physical experiments. However, once the workload drivers have been specified, they should remain consistent with and without technology inclusion. If available, physical experiment data can be utilized to benchmark the correlation between workload and changes to the operational model's environment, similar to the results from Experiment 1. Performance assessments are similar to those currently conducted in systems analysis, however, discrete tasking between the operator and the technology enables the analysis of performance variations due changes in the execution of shared responsibility. Further, the continuous measurement of the workload metrics enables of sectional analysis, looking for periods of undesirably high or low workloads. This is shown in the autopilot example where the operator's workload to monitor the automation remains continually high. Similarly, the breakdown in actions allows an understanding of the operator's involvement in the task. A lack of operator involvement could show signs of disengagement leading to a decrease in awareness.

The process diagram for the methodology is shown in Figure 5.2. Since this methodology is intended as an addition to the current OR process, this figure shows how the new model components interact with the original 7-steps. The steps are shown on the left while the dashed lines shows the horizontal correlation of new tasks required in each step to de-

fine and utilize methodology components. Starting with the orientation step, it remains the same, still focusing on understanding the operational context. Problem definition must be expanded. For the operations model, a system-level concept of operations (CONOPs) is created. However, the additional model components require further definition regarding the role of the operator and desired automation technologies. The CONOPs is used to define the operator's overall goals for the dynamic task list and the elements of the operator's awareness for the sensory model. Additionally, workload drivers are determined based on experimental findings that are contextually relevant to the current CONOPs. The data collection step requires an HTA breakdown of the goals from the problem definition until a fidelity is met that specifies the operator's interaction with the system. These can be found in literature or created using subject matter expertise. Additionally in this step, awareness attributes (ex: time to check a sensor) are researched or determined through SA studies. The data pieces are then utilized in the formulation step to define an operator-focused agent. The goals from the HTA breakdown are mapped to information-decision-action sequences that can be utilized to create an operator-focused behavior tree. The awareness attributes are used to specify the operator's information acquisition and memory capabilities, ensuring that the operator is acting on information that he would likely have available and not the system capabilities. With the sensory model and dynamic task list defined, the operator's actions and awareness elements are mapped to workload metrics based on the literature search conducted during problem definition. With all four model components defined, the model solution provides additional output metrics for the operator's actions, awareness, and workload throughout the scenario. These outputs are utilized for validation, specifically ensuring alignment between the virtual workload variations and expected variations. If the metric is determined sufficient, then proceed back to model formulation for the introduction of each technology. Technologies vary the behavior tree splitting tasks between the automation and the operator based on the CONOPs. Technology also impacts the awareness model, changing the data monitored by the operator. After each technology change,

the outputs should be analyzed to ensure the benefits and challenges of the automation are being captured and the data variation are inline with expectations. With workload data sets created for each technology, the information can be combined to create an automation tradespace capable of assessing the role of the operator. This tradespace can compares the actions and dynamic workload's for the operator throughout the CONOPs scenario looking for undesirable peaks, averages, or valleys. These insights were previously not allowed when assessing at the system level.

The positive impact of the methodology has been demonstrated in the two technology case studies above (driving technologies and manned-unmanned teaming for an ISR mission). Together they show how implementing this methodology enables operator insights and technology evaluations. The traffic model is a more detailed implementation of the methodology. The availability of information in the domain allowed for the capture of the focus agent, and the relative homogeneity of a basic driving environment enable an evaluation with fewer simplifications. The verification of the methodology was broken apart into separate experiments. Experiment 1 baselines the variations created from implementing the new virtual workload methodology in two operations models (highway and city) to physical experiment data found in literature. Experiment 2, focusing on the highway operations model, captures the changes required in the dynamic task list and sensory model for technology inclusion. Experiment 3 demonstrates how the new workload data can be utilized as a dimension of a systems tradespace. The methodology also captures the detailed operator's actions which provides a different metric for possible over-loading and under-loading conditions. The low flow condition in Figure 3.15b can show under-loading of the operator, because he spends 62% of his time at a mental workload of 0 with a maximum duration of 55 seconds between actions. The time spent at different workload levels and between actions is a new tradeoff metric not typically associated with the single-point NASA-TLX scores. Experiments 1-3 break apart the methodology, in contrast, Experiment 4 repeats all four steps of the methodology again together to demonstrate the applicability within

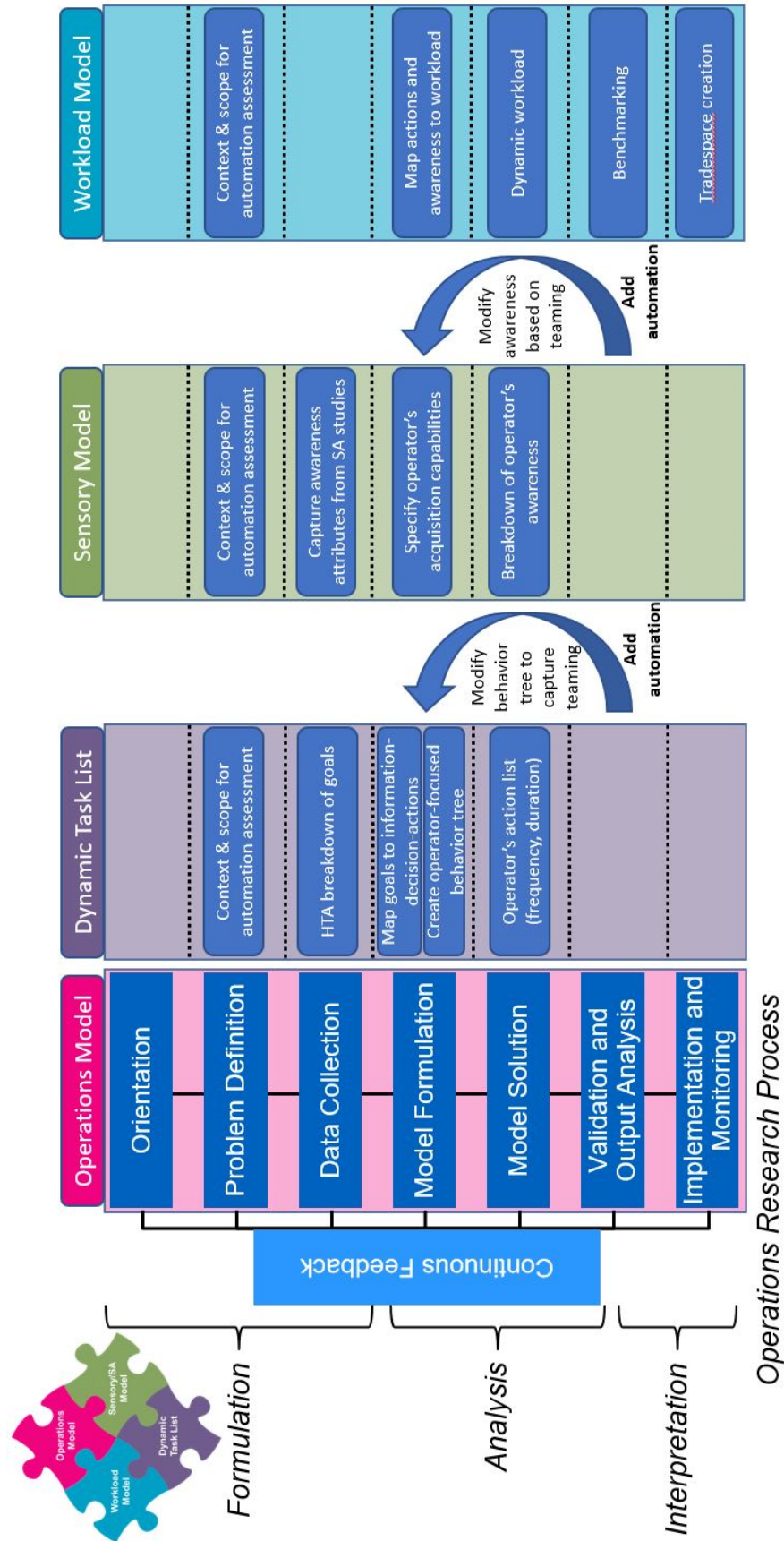


Figure 5.2: Process for capturing the role of the operator in a virtual environment to enable automation technology assessments

the aerospace domain. An ISR mission is performed independently by a piloted aircraft, then the same mission is performed with the help of a UCAV. The results demonstrated an expected decrease in workload by the pilot as he switches to UCAV management instead of primary actor, however, the results also show a variability in performance based on the UCAV's capabilities and timing of management tasks.

The two case studies captures some of the difficulties expected during the implementation of an operator-focused methodology. The first challenge is the expected variability in the environments that the technology is expected to operate. The traffic model captures a basic scenario, but it is representative of many highway layouts worldwide. The highway system is somewhat rigidly defined and the a single operations model can provide considerable insight. Conversely, the ISR mission is dependent on the red laydown and checkpoint locations, and multiple laydowns should be defined and analyzed to fully capture the impact of the UCAV's integration. Along these lines, another challenge of this methodology is the operator data required. Improvements to the focus agent are dependent on the availability of environment laydowns for the operations model and operator data for the dynamic task list, sensory model, and the workload model. The data exists in the driving domain, but it is publicly lacking in the military aerospace domain. Therefore, the methodology requires data collection through physical studies or consultations with subject matter experts. Unlike conducting each technology evaluation study with operators-in-the-loop, the data collected is focused on defining the operator thus it can be utilized to create multiple virtual studies reducing the overall study costs and timelines. Another challenge, as with any increase in model fidelity, is the increase in runtime due to the additional model elements and details. This methodology requires more background elements to be captured and data processing for both the operator and the system. The traffic model showed the biggest runtime impact due to the number of agents. To create traffic jams and other features expected on the road, all agents must have some level of awareness. Additionally, the density of elements in the operator's vision and the rate that they enter and leave memory impacted the runtime. Over-

all, the high-density traffic model took 20 minutes of processor time to run 10 minutes of simulation time, compared to the 5 seconds taken to run the 2 hour ISR mission. Changing modeling techniques, number of elements modeled, and modeling language will all impact the runtime. Overall, the 20 minutes is still quicker and cheaper than conducting a physical experiment, however, care should be taken to start with essential elements in the operations model to properly manage runtime. Lastly, the two trade studies only provide an insight into a full automation technology assessment. The driving study could look at multiple weather conditions, if it is expected to operate in multiple conditions. This would effect the desired operating speed in the dynamic task list and the sensory model's sight distance and time to acquire data. Similarly, the ISR mission could also add weather to change the line of sight conditions or change the mission conditions such as having unknown radar locations and requiring real time adjusts as the radar's locations are determined. Alternative control teaming methods can also be analyzed, such as reducing the lull in the UCAV's mission by splitting management between two piloted aircraft in the CAP. The basic case studies help validate the continued development of the methodology. Although, there are challenges to the operator-focused methodology, the reduction in uncertainty is a key enabler for automation technology evaluations.

This methodology provides the first step in decreasing the uncertainty associated with assessing automation technology inclusion early in the design process. The uncertainty is reduced by separating the automation from the operator and explicitly defining task assignments and awareness considerations within the operations assessment. It does not discuss the creation of the entire system tradespace, but adds workload metrics as a dimension to the existing space. Although AFSIM is utilized during the implementation of both case studies, the steps are applicable to any agent-based modeling language. The steps breakdown an improved way of defining agent elements to capture the teaming between the operator and automation technologies. The detailed breakdown utilized by the methodology forces early decisions on defining the interaction between the automation and the operator.

The tradespaces created highlight the strengths and weaknesses of automation technology inclusion.

5.1 Contributions

This work creates a methodology that reduces the uncertainty experienced in performance evaluations of increasingly automated systems by capturing the role of the operator and resulting workloads. The role of the operator has been identified as an essential pillar for increased autonomy for aircraft systems. The introduction of automation changes the role of the operator, and plays a direct role on the performance of the system. An over- or under-stimulated operator will decrease overall system performance or worse they can cause system failure. The relative immaturity and closed nature of automation in the aviation industry motivated searches outside the industry for current automation challenges and benchmarking data. A study on air traffic controllers showed full automation reduced the role of the controller too much effecting performance, and in the crashes of Tesla Autopilot the National Transportation Safety Board has partially blamed on a design flaw by allowing the operator too much disengagement from the driving task. Three gaps were identified in current operations models concerning the role of the operator that are increasing the uncertainty for partially automated system analysis. The first gap was current agent-based models focus on the system as a whole, without capturing any effects between the human-machine. The second is that the operator is assumed omnipresent and modeled based on system capabilities. Third, operator workload, when captured, is set as an input and not measured as an output, thus it cannot be viewed or assessed early in the design process. This methodology solves these gaps by including a dynamic task list, sensory model, and workload assessment metric into the operations model. These additions allow operator workload to move from a static input to a dynamic output, which can be assessed across a technology trade space and be utilized by the decision maker to specify requirements and select technologies. The workload can now be assessed across the operational scenario or

regions or high and low workload can be identified and quantified. As technology trends towards autonomy, reducing design uncertainty in the development of partial automation becomes of high importance.

The primary contribution of this new methodology is reducing uncertainty around the operator's workload and awareness in partially automated systems. Dynamic and traceable workload metrics for mental and temporal loads can now be created virtually. The metrics showed similar trends and magnitude differences as those conducted experimentally and found in literature. The detailed breakdown of tasking between the operator and automation also ensures a more representative performance metric for system analysis. The performance is now tied to the operator's awareness and actions, instead of the system's capabilities. This becomes important as the system's tasks and awareness responsibilities are shared between the operator and the automation. In order to get an accurate technology assessment during the early development phases, the role of the operator and impact of the technology must be more explicitly modeled. Through the inclusion of Cognitive Engineering disciplines into the Operations Research and Systems Engineering fields, an agent-based model can be created to dynamically monitor operator workload and system performance. This dynamic measurement can then be utilized in the requirements definition and TIES processes by the decision maker. This methodology provides the means by which systems like Boeing's Airpower Teaming System may be evaluated and compared to traditionally piloted aircraft.

This research project creates a methodology capable of operator workload quantification during operations assessments of partially automated systems. The requirements for this methodology (as discussed above) are: dynamic, quantifiable, re-configurable, and traceable. The first three experiments prove out the components of the methodology utilizing an automotive example due to data availability. Experiment 1 is focused on dynamic and quantifiable, Experiment 2 is focused on re-configurable, and Experiment 3 shows the traceability between the measured workloads and integrated technologies/environmental

factors. Experiment 4 demonstrates the entire methodology within the aerospace domain by applying it to evaluate manned-unmanned teaming. The results shows that adding a dynamic task list, sensory model, and workload model to an operations model enables the evaluation of the human's role during automation technology integration.

5.2 Future Work

This work is envisioned as a starting point for operations assessments of partially automated systems. ASTM International identified six pillars that are critical for increased autonomy in aircraft systems [23]. This work focused on operational considerations and the human role, however, operations analysis and this methodology could be expanded to account for other pillars. The two identified areas of future work are: 1) using this methodology to assess different management techniques for automation failure modes; and 2) expanding the elements captured by the workload metrics. The pillars dictate that critical automation systems must have run-time assurance to ensure that the automation is designed to catch mistakes and recover from them to maintain safe flight [66]. The methodology can be utilized to assess the impact of a system being online or offline on the operator's workload. The UCAV case study demonstrates the idle time experienced if the operator is unavailable based on the specified management style. Currently the UCAV just loiters, however, this lack of oversight could also result in undesirable levels of pilot workload depending on which automation functions are supposed to be monitored and how they fail. Examples of system failure are seen in the highway autopilot which can struggle with poor visibility from weather, bright lights, interference for the ultrasonic sensors, or obstructions to the camera [5]. With the inclusion of technology failure rates and modes, the methodology can be expanded to understand critical levels of operator awareness and which tasks the operator must prioritize. The dynamic task list would require modifications for these interventions due to failure: how the operator determines failure and what the steps are for failure. The failure branch is separate from the monitoring and normal correction or

management branch, because it is rushed and may happen at a different awareness level. Upon takeover, the operator enters the technology failure branch. This branch represents the tasks for manual system control, however, the operator may have degraded performance due to lack of practice/involvement when the automation was active. This performance degrade will manifest in longer times to execute certain tasks as shown by the ATC study [27]. The operator's SA and associated tasking also changes based on the failure branch's specification instead of the SA dictated by the monitoring branch.

The second area for future work concerns the workload metrics. Both trade studies focus on one element in the operator's memory for the workload metrics. Although this approach creates a good starting point and aligned well to physical experiment data for the driving case study, further work can be done on combining and expanding workload metrics. Human-machine interfaces have been extensively studied to understand how the pilot's workload is effected by different data presentation techniques [3, 92]. This level of fidelity is currently not captured by the workload methodology, because it is intended to be utilized during early design analysis. As design decisions are made and the tradespace is narrowed, information layout characteristics may be analyzed in an operational context by expanding this methodology. If the methodology is expanded to include operator workloads in the cockpit, then an additional focus should be placed on combining different workload types. The current methodology assumes that the mental and temporal demand due to maneuvering encompasses the evaluation of the flight deck, however, evaluating flight deck layouts requires workload types to be separated. Individual workload metrics must be defined for each operator action and awareness element. Although this breakdown in workloads may help facilitate discussion on the different drivers of workload, a utility function capable of combining these disparate values is required to enable the analysis of the operator's composite workload.

Appendices

APPENDIX A

SUMMARY OF RESEARCH QUESTIONS

Table A.1: A summary of research questions with related sub-questions and assertions/hypotheses

RQ	
1.0	<p>Can the inclusion of cognitive engineering disciplines in the operational assessment of systems reduce the uncertainty pertaining to the operator's cognitive load by providing a dynamic, traceable, and quantifiable measurement throughout the scenario?</p> <p>Hypothesis: the inclusion of a dynamic task list, SA model, and workload model in an agent-based operations model (AFSIM) will enable a dynamic, traceable, and quantifiable operator workload measurement similar to those currently manually conducted.</p>
1.1	<p>Can a dynamic task list be created from the operator's perspective to map operator awareness to a goal-action agent-based rule set?</p> <p>Assertion: utilizing HTA and representing goal-action sequences will enable the creation of a dynamic task list that can create an agent rule-set capable of guiding behaviors based on operator awareness</p>
1.2	<p>Can the operator's knowledge acquisition and information utilization processes be explicitly modeled to include acquisition delays and utilization challenges which directly impacts awareness and how the agent follows its dynamic rule set?</p> <p>Assertion: the studies conducted in the field of SA will allow for the dynamic representation of the costs of information acquisition and the challenges with information utilization which can be utilized in the activation sequence of the dynamic task list</p>
1.3	<p>Can the typically manual, human-based workload measurement techniques be applied in an agent-based environment to provide a quantitative assessment of workload?</p> <p>Hypothesis: the detailed task list will map to different elements of the workload measurement techniques providing a dynamic, traceable assessment throughout the simulation</p>

Table A.2: A summary of research questions with related sub-questions and assertions/hypotheses (cont.)

RQ	
2.0	<p>How should variations in the system due to technology integration be captured within a modeling environment to properly capture the effect on the system's performance and enable the assessment of changes to the operator's cognitive load?</p> <p>Hypothesis: The modifications, additional three branches on the dynamic task list and variations to SA according the operator trust, will allow technology trade space exploration based on operator workload and system performance.</p>
2.1	<p>What changes must be done to the agent's mental model to capture changes in technology or automation of the system?</p> <p>Assertion: Adding monitoring, correction, and failure branches to the dynamic task list will modify the agent's behavior and task load differing amounts based on the technology integrated.</p>
2.2	<p>How should the operator's trust be captured within the simulation such that the Situational Awareness and task loading of the operator are adjusted based on the level of trust in the automation?</p> <p>Hypothesis: By changing the duration which data is stored in the operator's SA model (if also monitored by the automation) and the frequency of monitoring tasks based on the level of trust in the technology, it will approximate the performance differences shown by different skill-level operators.</p>
3.0	<p>What effect will reducing the uncertainty around the operator's cognitive load, by enabling the quantification and identification of the cognitive factors, have on performance evaluations and technology integration decisions?</p> <p>Hypothesis: Reducing uncertainty concerning cognitive load will allow decision makers to locate possible failure points due to: peaks, valleys, or overall high task loading, or inadequate situational awareness.</p>
4.0	<p>Can the operator-focused methodology provide insights to a current area of focus for the aerospace domain through the explicit modeling of the piloted aircraft and the automation technology?</p> <p>Hypothesis: The operator-focused methodology is capable of decreasing uncertainty in the analysis of manned-unmanned teaming.</p>

APPENDIX B

SUMMARY OF EXPERIMENTS

Table B.1: A summary of experiments

Experiment 1:	
Objective	Compare the results and general trends of an agent-based model to a study using the NASA TLX scale and participants
Setup	Create an agent-based model using AFSIM to assess the cognitive load associated with driving in different scenarios - The dynamic task list and SA model will be built out to represent the driving activity - The goal of the operator (driver) will be traversing between randomly assigned start and end points
Output	Workload variations across the 3 TLX (mental demand, temporal demand, performance) scales based on changes to scenario (highway and city) but not varying the agent's mental model
Experiment 2:	
Objective	Compare the dynamically calculated workload and performance changes after automation integration with studies done for automated cars
Setup	Modify agent-based model from Experiment 1 Add technologies: - Adaptive cruise control - Highway autopilot Modify the mental model: - Task sequencing must be changed to include variations introduced by the technology - Operator trust level must be specified as an input to determine level of knowledge overlap and workload
Output	Compare cognitive workload changes with those of studies done for automated cars

Table B.2: A summary of experiments (cont.)

Experiment 3:	
Objective	Assess the capabilities of the operator workload measurement framework
Setup	Analyze the simulation outputs from Experiment 2: - Create an interface for decision makers to understand operator workload throughout the mission - Graph the data being kept in the Operator's awareness model throughout the simulation. - Enable the comparison of different system technologies
Output	Compare the outputs of the dynamic driving assessment to the Tesla studies on driver attention and workload prior to accidents
Experiment 4:	
Objective	Test the methodology on a use case that is of growing interest and a primary motivation of this work: manned-unmanned teaming
Setup	Create an agent-based model for an intelligence, surveillance, reconnaissance (ISR) mission: Capture workload assessment scores throughout the mission: - Mental Demand - Temporal Demand - Performance Integrate technology : Unmanned Wingmen - Change the dynamic task list and sensory model Build a technology trade space with pilot cognitive factors as one of the metrics
Output	Workload measurements throughout the mission duration across the 3 workload scales A trade space showing the effects of manned/unmanned teaming

REFERENCES

- [1] N. R. C. 1998, “Modeling Human and Organizational Behavior: Application to Military Simulations,” 1998.
- [2] “Aerospace and Defense Simulation and Scheduling Software,” 2019.
- [3] A Aiello *et al.*, “Run-Time Assurance for Safety-Critical Flight Control Systems,” 2010.
- [4] “Airport Surveillance Radar (ASR-11),” 2020.
- [5] “Autopilot and Full Self-Driving Capability.” (2021), (visited on 05/18/2021).
- [6] A. T. Bahill and F. F. Dean, “Discovering System Requirements, Chapter 4,” *Handbook of Systems Engineering*, vol. 18(1), pp. 205–266, 2009.
- [7] V. Banks and N. Stanton, “Keep the driver in control: Automating automobiles of the future,” vol. 53, 2016.
- [8] J. J. Bestard, “Air Force Research Laboratory Innovation – Pushing the Envelope in Analytical Wargaming,” *Journal of Cyber Security and Information Systems*, vol. 4(3), pp. 12–17, 2016.
- [9] E. Bonabeau, “Agent-based modeling: Methods and techniques for simulating human systems,” vol. 99(suppl3), 2002.
- [10] B. Castello, “The Operations Research Problem Solving Process,” 2010.
- [11] K.-S. Chang and D. Zhu, “2.4 hierarchical finite state machine (hfsm) & behavior tree (bt),” 2005.
- [12] J. Y. C. Chen, “Concurrent Performance of Military and Robotics Tasks and Effects of Cueing in a Simulated Multi-Tasking Environment,” *Presence: Teleoperators and Virtual Environments*, vol. 18(1), pp. 1–15, 2009.
- [13] J. Y. C. Chen and M. J. Barnes, “Robotics Operator Performance in a Military Multi-Tasking Environment,” 2007.
- [14] A. Chialastri, “Automation in Aviation,” *Automation*, 2012.
- [15] G. Chruch, “Aerospace Moving Forward with Modeling and Simulation,” 2019.

- [16] N. A. of City Transportation Officials, “Urban Street Design Guide,” 2020.
- [17] P. D. Clive, J. A. Johnson, M. J. Moss, J. M. Zeh, B. M. Birkmire, and D. D. Hodson, “Advanced Framework for Simulation, Integration and Modeling (AFSIM),” *International Conference Scientific Computing*, 2015.
- [18] D. Coldeway, “Boeing’s ‘Wingman’ drone buddies up with pilot-flown jets,” 2019.
- [19] O. contributors, “OpenStreetMap,” 2021.
- [20] N. J. Cooke and F. Durso, “Stories of Modern Technology Failures and Cognitive Engineering Successes,” 2007.
- [21] D. Delaney, K. Klockner, and Y. Toft, “Assessing the Human Factor in Truck Driving,” *Safety Engineering Series*, vol. 14(1), pp. 1–13, 2019.
- [22] “Department of Defense Handbook - System Requirments Document Guidance,” vol. MIL-HDBK-520A, 2011.
- [23] “Developmental Pillars of Increased Autonomy for Aircraft Systems,” 2020.
- [24] C. Dickerson and D. Mavris, “A Brief History of Models and Model Based Systems Engineering and the Case for Relational Orientation,” *IEEE Systems Journal*, vol. 7(4), pp. 581–592, 2013.
- [25] C. E. Dickerson and D. N. Mavris, “Architecture and Principles of Systems Engineering,” 2009.
- [26] *e-Assist*, <https://www.mitsubishi-motors.com/en/innovation/technology/library/e-assist.html?intcid=technology-library-e-assist>, Accessed: 11-20-2019.
- [27] T. Edwards, J. Homola, J. Mercer, and L. Claudatos, “Multifactor interactions and the air traffic controller: The interation of situation awareness and workload in association with automation,” *International Federation of Automatic Control (IFAC)*, pp. 597–602, 2016.
- [28] M. Endsley, “Measurement of Situation Awareness in Dynamic Systems,” *Human Factors*, vol. 37(1), pp. 65–84, 1995.
- [29] M. R. Endsley, “Design and evaluation for situation awareness enhancement,” *Proceedings of the Human Factors Society annual meeting*, vol. 32(2), 1988.
- [30] ———, “A Survey of Situation Awareness Requirements in Air-to-Air Combat Fighters,” *International Journal of Aviation Psychology*, 1993.

- [31] ———, “Toward a Theory of Situation Awareness in Dynamic Systems,” *Human Factors*, vol. 37(1), pp. 32–64, 1995.
- [32] ———, “Situation Awareness Misconceptions and Misunderstandings,” *Journal of Cognitive Engineering and Decision Making*, vol. 9(1), pp. 4–32, 2015.
- [33] “F-16 Combat Aircraft Fundamentals,” *Multi-Command Handbook 11-F16*, 1996.
- [34] “Fast-tracking Safe Autonomous Vehicles,” *White Paper*, 2018.
- [35] C. Fox and G. Ülkümen, “Distinguishing two dimensions of uncertainty,” 2011.
- [36] K. Fox, “Future Direction: Increased Autonomy,” *Connecting the Dots in Cybersecurity and UAS*, 2014.
- [37] M. Goolsbay, *USAFE force structure analysis*, Conference Presentation, 87th MORS Symposium, Jun. 19, 2019.
- [38] I. Grigoryev, “AnyLogic 7 in Three Days,” 2016.
- [39] ———, “AnyLogic Is The Technology And Market Leader In Simulation Tools For Business Applications,” 2016.
- [40] M. Guckenberger, *Modeling the interplay between capabilities and tactics to reach enhanced system of systems performance*, Conference Presentation, 87th MORS Symposium, Jun. 19, 2019.
- [41] R. P. Haffa Jr. and A. Datla, “Joint Intelligence, Surveillance, and Reconnaissance in Contested Airspace,” *Air & Space Power Journal*, vol. 28(3), 2014.
- [42] D. H. Harper, “Operations Analysis Integration for Effectiveness-Based Design in the AFRL EXPEDITE Program,” *AIAA Scitech 2020 Forum*, 2020.
- [43] C. M. Harris, M. D. R. Sokollek, L. S. Nunez, J. T. Valco, M. Balchanos, and D. N. Mavris, “Simulation-based UAS Swarm Selection for Monitoring and Detection of Migrant Border Crossings,” 2018.
- [44] D. Harris, “IMPRINT Pro User Guide,” vol. 1, 2009.
- [45] ———, “Engineering Psychology and Cognitive Ergonomics,” 2011.
- [46] D. Harris, N. A. Stanton, and A. Starr, “Spot the difference: Operational event sequence diagrams as a formal method for work allocation in the development of single-pilot operations for commercial aircraft,” *Ergonomics*, vol. 58(11), pp. 1773–1791, 2015.

- [47] S. Hart and L. Staveland, “Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research,” *Frontiers in Human Neuroscience*, 1988.
- [48] S. Hawking, “A Brief History of Time,” 1988.
- [49] P. T. Hester and A. Tolk, “Applying methods of the m&s spectrum for complex systems engineering,” 2010.
- [50] I. Hlaváček, J. Chleboun, and I. Babuška, “Chapter I. Reality, Mathematics, and Computation,” *North-Holland Series in Applied Mathematics and Mechanics*, vol. 46, pp. 1–49, 2004.
- [51] E. Hollnagel and D. D. Woods, “Joint Cognitive Systems: Foundations of Cognitive Systems Engineering,” 2005.
- [52] B. L. Hooey, D. B. Kaber, J. A. Adams, T. W. Fong, and B. F. Gore, “The Underpinnings of Workload in Unmanned Vehicle Systems,” *IEEE Transactions on Human-Machine Systems*, vol. 48(5), pp. 452–467, 2018.
- [53] D. Isla, “Gdc 2005 proceeding: Handling complexity in the halo 2 ai,” 2005.
- [54] N. Kalra and S. M. Paddock, “Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?” *Transportation Research Part A: Policy and Practice*, vol. 94, pp. 182–193, 2016.
- [55] A. Kaslikowski, “Everything you need to know about autonomous vehicles,” 2019.
- [56] S. S. Kirschenbaum, “Uncertainty and Automation,” 2003.
- [57] B. Kirwan and L. K. Ainsworth, “A Guide to Task Analysis,” 1992.
- [58] S. B. Klein, “Motivation: Biosocial approaches,” 1982.
- [59] G. Klien, D. D. Woods, J. M. Bradshaw, R. R. Hoffman, and P. J. Feltovich, “Ten challenges for making automation a ”team player” in joint human-agent activity,” *IEEE Intelligent Systems*, vol. 19(6), pp. 91–96, 2004.
- [60] M. Lacabanne, F. Amadiou, A. Tricot, and N. Spanghero-Gaillard, “Analysis of the flight task around different types of aircraft,” 2012.
- [61] R. J. Lay, J. W. Taylor Jr., and G. Brunins, “ARSR-4: Unique Solutions to Long-Recognized Radar Problems,” *IEEE International Radar Conference*, 1990.
- [62] S. E. Lee, E. C. Olsen, and W. W. Wierwille, “A Comprehensive Examination of Naturalistic Lane-Changes,” *Ergonomics*, vol. 58(11), pp. 1773–1791, 2015.

- [63] J.-H. Lewe, B.-H. Ahn, D. P. Schrage, D. A. DeLaurentis, and D. N. Mavris, “An Integrated Decision-Making Method To Identify Design Requirements Through Agent-Based Simulation For Personal Air Vehicle System,” 2002.
- [64] W.-C. Li, “The Analysis of Human Errors in a Large Commercial Aircraft when Performing a Go-around,” *Journal of Aeronautics, Astronautics, and Aviation*, vol. 42(1), pp. 37–46, 2010.
- [65] W.-C. Li, F.-C. Chiu, Y.-s. Kuo, and K.-J. Wu, “The Investigation of Visual Attention and Workload by Experts and Novices in the Cockpit,” *International Conference on Engineering Psychology and Cognitive Ergonomics*, 2013.
- [66] Y. Lim *et al.*, “Avionics Human-Machine Interfaces and Interactions for Manned and Unmanned Aircraft,” *Progress in Aerospace Sciences*, vol. 102, 2018.
- [67] M. K. Lotov A.V., “Visualizing the Pareto Frontier,” *Branke J., Deb K., Miettinen K., Słowiński R. (eds) Multiobjective Optimization. Lecture Notes in Computer Science*, vol 5252., 2008.
- [68] G. Marsden, M. McDonald, and M. Brackstone, “Towards an understanding of adaptive cruise control,” vol. 9, 2001.
- [69] C. Martin, J. Cegarra1, and P. D. Averty, “Analysis of Mental Workload during En-route Air Traffic Control Task Execution Based on Eye-Tracking Technique,” *Engineering Psychology and Cognitive Ergonomics*, pp. 592–597, 2011.
- [70] I. F. Martín, “Valuation of Design Adaptability in Aerospace Systems,” 2008.
- [71] D. N. Mavris and D. A. DeLaurentis, “Methodology for examining the simultaneous impact of requirements, vehicle characteristics, and technologies on military aircraft design,” 2000.
- [72] D. N. Mavris and M. R. Kirby, “Technology Identification, Evaluation, and Selection for Commercial Transport Aircraft,” 1999.
- [73] D. N. Mavris and A. Sudol, “Formulation And Implementation Of A Method For Technology Evaluation Of System Of Systems,” 2018.
- [74] “MBA-H2040: Quantitative Techniques for Managers - Introduction to Operations Management,” 2001.
- [75] R. McKendrick, B. Feest, A. Harwood, and B. Falcone, “Theories and Methods for Labeling Cognitive Workload: Classification and Transfer Learning,” *Frontiers in Human Neuroscience*, vol. 13, pp. 1–20, 2019.

- [76] “MIL-STD-3013: Glossary of Definitions, Ground Rules and Mission Profiles to Define Air Vehicle Performance Capability,” 2008.
- [77] “Mitigation Strategies For Design Exceptions - Lane Width,” 2014.
- [78] MITRE, “Technology, Planning and Integration for Observations (TPIO) Trade-Space Analysis Guide,” 2012.
- [79] D. Morales and M. L. Cummings, “UAVs as Tactical Wingmen: Control Methods and Pilots’ Perceptions,” *Unmanned Systems*, vol. 23(1), 2005.
- [80] P. M. Morse, G. E. Kimball, and S. I. Gass, “Methods of Operations Research,” 2003.
- [81] J. Paxion, E. Galy, and C. Berthelon, “Mental workload and driving,” *Frontiers in Psychology*, 2014.
- [82] D. Pegden, “An Introduction to Simio for Beginners,”
- [83] “Pre-Milestone A and Early-Phase Systems Engineering: A Retrospective Review and Benefits for Future Air Force Systems Acquisition,” *National Research Council of the National Academies*, 2008.
- [84] “Prospect distance from the ground and line of sight distance between two observers,” 2011.
- [85] N. T. S. B. O. of Public Affairs, “Driver Errors, Advanced Driver Assistance System Design, Led to Highway Crash,” 2019.
- [86] N. I. A. Rahman, S. Z. M. Dawal, and N Yusoff, “Subjective responses of mental workload during real time driving: A pilot field study,” *IOP Conference Series: Materials Science and Engineering*, vol. 210, 2017.
- [87] D. P. Raymer, “Aircraft Design: A Conceptual Approach (Fourth Edition),” 2006.
- [88] S. D. Roberts and D. Pegden, “History of Simulation Modeling,” 2017.
- [89] “SAE standards news: J3016 automated-driving graphic update,” 2019.
- [90] “Safety 1n Numb3rs - What You Can Do,” 2020.
- [91] P. Salmon, D. Jenkins, N. Stanton, and G. Walker, “Hierarchical task analysis vs. cognitive work analysis: comparison of theory, methodology and contribution to system design,” *Theoretical Issues in Ergonomics Science*, vol. 11(6), pp. 504–531, 2010.

- [92] J. Schierman, M. DeVore, N. Richards, and M. Clark, "Runtime Assurance for Autonomous Aerospace Systems," *Journal of Guidance, Control, and Dynamics*, vol. 42(12), 2020.
- [93] M. Schneider, J. McGrogan, J. M. Colombi, M. E. Miller, and D. S. Long, "Modeling Pilot Workload for Multi-Aircraft Control of an Unmanned Aircraft System," *INCOSE International Symposium*, vol. 21(1), pp. 796–810, 2011.
- [94] D. P. Schrage, "Technology for Rotorcraft Affordability Through Integrated Product/Process Development (IPPD)," 1999.
- [95] D. P. Schrage, D. A. DeLaurentis, and K. Taggart, "IPPD Concept Development Process for Future Combat System," 2002.
- [96] "Science and Technology Strategy - Strengthening USAF Science and Technology for 2030 and Beyond," 2019.
- [97] C. L. Scovel, "Perspectives on Overseeing the Safety of the U.S. Air Transportation System," 2019.
- [98] T. B. Sheridan and W. L. Verplank, "Human and Computer Control of Undersea Teleoperators," pp. 168–170, 1978.
- [99] C. R. Shrader, "History of Operations Research in the United States Army, Volume I: 1942-1962," 2006.
- [100] D. S. Soban and E. Upton, "Design of a UAV to Optimize Use of Fuel Cell Propulsion Technology," 2005.
- [101] N. Stanton and M. Young, "Driver behaviour with adaptive cruise control," *Ergonomics*, vol. 48, 2007.
- [102] N. A. Stanton, "Hierarchical task analysis: Developments, applications, and extensions," *Applied Ergonomics*, vol. 37, pp. 55–79, 2006.
- [103] J. Stillion, "Trends in Air-to-Air - Combat Implications for Future Air Superiority," 2015.
- [104] N. Strout, "Air Force wants to expand tactical data network to space," 2019.
- [105] A. Sudol, S. Edwards, and D. Mavris, "Technology Assessment for Launch Vehicle Concepts Using The DYnamic Rocket Equation Tool (DYREQT)," 2018.
- [106] "Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities," vol. Version 3, 2006.

- [107] V. Tabora, “Tesla Enhanced Autopilot Overview — L2 Self Driving HW2,” 2018.
- [108] K. H. Teigen, “Yerkes-Dodson: A Law for all Seasons,” *Theory & Psychology*, vol. 4(4), pp. 525–547, 1994.
- [109] “Tesla Autopilot.” (2020), (visited on 12/18/2019).
- [110] “The Autonomy Paradox.” (2011).
- [111] D. Thomas and D. Emerson, *AFSIM Training*, <https://www.infoscitex.com/afsim-training/>, Accessed: 01-20-2020.
- [112] A. Tolk, “Merging two worlds: Agent-based simulation methods for autonomous systems,” *Modeling, Simulation & Visualization Engineering Faculty Publications*, 2015.
- [113] “US Code - Title 10, part I,” 2016.
- [114] D. Westerlund, “How To Deal With Cognitive Load In UX and Voice Design,” *IOP Conference Series: Materials Science and Engineering*, vol. 210, 2017.
- [115] U. Wilensky, “NetLogo User Manual, version 6.1.1,” 2019.
- [116] J. C. de Winter, R. Happee, M. H. Martens, and N. A. Stanton, “Effects of adaptive cruise control and highly automated driving on workload and situation awareness: A review of the empirical evidence,” *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 27, 2014.
- [117] C. Woodyard, “On autopilot: ‘Pilots are losing their basic flying skills,’ some fear after Boeing 737 Max crashes,” 2019.
- [118] G. N. Yannakakis and J. Togelius, “Artificial intelligence and games,” 2018.
- [119] C. J. Yi, “Manned-Unmanned Teaming—An Analysis Of UAVs And Their Interoperability With Manned Aircraft,” *Journal of Singapore Armed Forces*, vol. 42(1), 2016.
- [120] C.-S. Y. Yu, E. Min-Yang Wang, W.-C. Li, and G. Braithwaite, “Pilots’ visual scan patterns and situation awareness in flight operations,” *Aerospace Medicine and Human Performance*, vol. 85(7), 2014.
- [121] K. B. Zandin, “Maynard’s Industrial Engineering Handbook,” 2001.
- [122] Y. Zhang and T. Kumada, “Relationship between workload and mind-wandering in simulated driving,” *PLoS ONE*, vol. 12, 2017.